Assignment 2: Malware Analysis

Your assignment is on Malware Analysis, which is based on the Lectures and Previous labs. The assignment is an individual assignment and is worth 30% of the module marking. You will be assessed on your ability to carry out a successful memory forensics investigation and report the artefacts and malicious activities analysed.

Marks Breakdown

You will be given a malware sample and an infected memory dump and set the task of analysing these. You are required to submit a forensics investigation report on your findings. To help you with constructing your report, Task-1 guides you to cover the main points that should be included in the report. Once you have completed the task you need to submit one report that contains the results of your investigation in PDF format.

30 Marks For all tasks Which is broken down into:

- **3 Marks:** For clarity of your description.
- **27 Marks:** For Task-1 question (breakdown below)

Experimental Setup

We will use the same setup of Lab 2 (Memory Forensics) to work with Volatility and Analyse the memory samples below.

You can use any instance of the Lab 2 (Memory Forensics) VM you already have or you can create a new one. To create a new one pick or create a folder. Then on a university computer right-click the folder and select "DOS shell" and then "Vagrant". If you are using your own computer then right-click and select terminal. Once you have a command prompt you can:

```
git clone https://git.soton.ac.uk/rht1g21/Lab2.git
```

Task-1

You've been given a malware sample and an infected memory dump. The malware is allegedly part of a high-scale APT attack. The antivirus industry is calling it "jackal," but detailed information is currently scarce. You searched Twitter and found someone saying "jackal's c2 list is just base64 and xor" but he didn't provide any hashes so you're not even sure if he's talking about the same executable.

```
wget https://git.soton.ac.uk/rht1g21/jackall2/-/raw/main/jackal.exe.zip
unzip jackal.exe.zip jackal.exe password: infected

wget https://git.soton.ac.uk/rht1g21/jackall2/-/raw/main/Vmem/jackal.vmem.7z
sudo apt-get install p7zip-full

7za e jackal.vmem.7z
```

1. What is the relevant profile to be used to analyse the provided memory image? (1 Mark)

   Answer:
   Win7SP1x86_23418

2. Obtain an unpacked sample of the malware. Specifically, use procdump with and without the –memory option. Are the two output files the same? Why or why not? (4 Mark)

   Answer:
   No, the two output files are not the same.

   Key differences:

   Without `-memory` option:

```
volatility -f jackal.vmem --profile=Win7SP1x86_23418 procdump -p 3028 -D .
```

   - It's MD5 checksum is as follows:
     22b0a433375b9ea2bb5482118df8f8de
   - Extractsthe on-disk executable (packed)
   - Contains UPX packing signatures (`UPX0`, `UPX1`)
   - Shows only basic API imports and minimal readable strings

   With `-memory` option:

```
volatility -f jackal.vmem --profile=Win7SP1x86_23418 procdump --memory -p 3028 -D .
```

   - It's MD5 checksum is as follows:
     60a06e27f324572efaa4892eb692085c
   - Captures the in-memory unpacked version
   - Reveals actual malicious functionality:
     - C2 commands (*exec*, *wget*)
     - Network operations (`URLDownloadToFileA`)
     - Process manipulation (`killproc`)

   The difference occurs because malware typically unpacks itself during execution. While the disk version remains packed to evade detection, the memory dump exposes the true malicious behavior, making it more detectable by security tools.

3. Analyse strings in the unpacked files. Make sure to use the -a flag to search the entire file and also check Unicode strings with -el. `strings -a FILENAME strings -a -el FILENAME` (3 Mark)

   Answer:
   The unpacked malware file contains clear indicators of malicious behavior:

   - **C2 Communication:** Use of APIs such as `URLDownloadToFileA` and `HttpOpenRequestA`, along with a custom User-Agent string: `The Jackal v4.2001`.
   - **Persistence:** Reference to a path: `c:\windows\system32\jackal.exe`, and a mutex named `__Dassara__`.
   - **Encoding:** Multiple Base64-encoded strings (e.g., `JyM9IiM9ISAqPSEiKzx5emBnPXlg`) were found, aligning with external reports suggesting the use of XOR encoding techniques.

   Figure 1 confirms that the malware is packed when stored on disk, but reveals its true functionality once it is unpacked and executed in memory.

Evidence:

```
78OI
IYCD
0GF9
M65P
Q810
KYKG
5WYY
2LHL
FAU1
47SG
1AUR
IEUH
HNFY
MRRU
4H2N
WN33
DB1L
JyM9IiM9ISAqPSEiKzx5emBnPXlg
ISElPSEjIj0iJCA9JCE8e3x8d2R6fXg8RXJ9cHxmZXZhPGV6YHpnPGBmfn52YT17Z35/
ISMjPSInJz0mIz0hIiE8YXZgcn5jf3Z3PHF2dXxhdj17Z35/
ISEiPSIlJj0iJSc9JiU8Y2Z/f3FycHg8anZge3plcns9Z2tn
JyY9IiYiPSIrID0iJyo8eHJge3ZhPEt2YXxren10PXlg
ISEkPSEhPSImJD0mPGZgcjxgfHBwdmE9e2d+fw==
ISIrPSIiID0iJCs9JCI8ZHZyZ3t2YT17Z35/
IiQrPSsmPSIqIj0mITx2dmNhfH48YHZldmF2PXtnfn8=
IiMiPSogPSohPSIlJDx8YGd2fWB8YWo8ZHpweHZ3PHJ/fw==
IiA9KyE9IiYiPSEiJjx/fGNjdmE8d2Z+cXF2f389Z2tn
KiM9ICY9ISAnPSEnKzx8YXp2fWc8cXJ4dmFgPXtnfn8=
IiMmPSQiPSEgJD0iJTxgZmN2YTxwf3JxcXZhYD1rfn8=
ICE9ISc9IScrPSIkKzx0fHx3emB7PHV2f398ZD17Z35/
IiMrPSIiJj0qKj0rJTxqdn9/fGR7cn5+dmE8YXZjfGFnYA==
IiErPSIgJz0iJCU9IiElPHZremBnYDxDcmByd3Z9cj13fHA=
JSQ9KyY9IScrPSEmPGRyenVgPHlmfXw9e2d+fw==
ISMhPSIqPSIqJD0iICI8fHBnfHF2YTxgcmdmYX09eWA=
IiYqPSEgIT0iIyE9IiYrPGNyYGB6fXQ8YHxmYXB2PHt8fnY9cmBj
IiUlPSEjPSYgPSEiKjxRZmFxcn14PF9mcHp3PHlycHhgPXtnfn8=
IiclPSshPSQkPSYqPGAqIioiIjx4f2B5ciIiPHV6f2d2YT1na2c=
ISAgPSIkIT0iKyM9IiUgPGFyfXB7dmE8ZHp9d3xkYD1je2M=
ISYhPSEhKj0iKiA9ISEkPGBnfHB4PGdhcnd6fXQ9e2d+fw==
ISIhPScgPSInIz0iJiE8YGN8YWc8e3Jje3JpcmF3PXlg
IisnPScrPSInID0iIiQ8ZHJ6Z3p9dD1yYGM=
ISYgPSEhJz0iJCI9ICo8fXZkYDxwfH5+PWN7Yw==
Jic9IiAqPSIrIz0iICs8YXxyYWA8ZXJhenxmdzxyf2d2YX1yZ3p8fT17Z35/
IiUrPSsnPSIqKz0hJys8YHZ2fTxqdn9jPXtnfn8=
IismPSEhKj0iJiQ9IiUrPHx9YXZwanZ7fDxldmFxcn89Z2tn
ISEqPSY9ISAgPSEjJDxndnpqajxgY39meD1yYGM=
IionPSEjKj0rKj0nIjxgdmF6fGA8e3J/cXZhPXtnfn8=
application/octet-stream
image/gif
text/*
__Dassara__
Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; The Jackal v4.2001
Software\Microsoft\Windows Player
>GET
c:\windows\system32\jackal.exe
vagrant@lab2:~$ strings -a -el executable.3028.exe
```

Figure 1: Partial output for `strings -a -el executable.3028.exe`

4. Based on the strings you see, describe the types of changes this malware may make to the running system's registry. Specifically, what key would you look for as an indicator of compromise? (1 Mark)

   Answer:
   The malware uses registry manipulation APIs including `RegCreateKeyExW`, `RegSetValueExW`, `RegEnumValueW` `RegQueryInfoKeyW` and `RegCloseKey`.

5. Was the malware actively accessing the key at the time of the memory dump? Note: you can specifically check for this using the handles plugin and filtering for open registry keys: (Use `volatility -f <FILE> handles -p PID --object-type=Key`) (3 Marks)

   Answer:

   Yes, the malware was actively accessing registry keys during the memory dump. Key evidence includes:

   Open Handles:

   - `\REGISTRY\USER\[ID]\Software\Microsoft\Windows Player`
   - `\REGISTRY\MACHINE\SOFTWARE\Microsoft\InternetSettings`

   These keys suggest the malware was modifying setting for persistence for Persistence/C2 communication. For a clear understanding Figure 2 shows terminal output screen.

```
vagrant@lab2:~$ volatility -f jackal.vmem --profile=Win7SP1x86_23418 handles -p 3028 --object-type=Key
Volatility Foundation Volatility Framework 2.6.1
Offset(V)     Pid     Handle     Access Type            Details
---------- ------ ---------- ---------- --------------- -------
0x920cf200   3028       0x18    0x20019 Key             MACHINE\SYSTEM\CONTROLSET001\CONTROL\NLS\SORTING\VERSIONS
0x8649eaf8   3028       0x20    0x20019 Key             MACHINE
0x85f7af50   3028       0x28        0x1 Key             MACHINE\SYSTEM\CONTROLSET001\CONTROL\SESSION MANAGER
0x937d3428   3028       0x78    0xf003f Key             USER\S-1-5-21-2833823845-3085568943-3082117713-1000
0x85f5a250   3028       0x7c    0xf003f Key             USER\S-1-5-21-2833823845-3085568943-3082117713-1000\SOFTWARE\MICROSOFT\WINDOWS PLAYER
0x9366e0f8   3028       0xa8    0x2001f Key             USER\S-1-5-21-2833823845-3085568943-3082117713-1000\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET SETTINGS
0x91436f60   3028       0xd4        0x1 Key             USER\S-1-5-21-2833823845-3085568943-3082117713-1000\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\EXPLORER
0x85fc5f28   3028      0x130    0x20019 Key             USER\S-1-5-21-2833823845-3085568943-3082117713-1000\SOFTWARE\POLICIES\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET SETTINGS
0x95b76260   3028      0x134    0x20019 Key             MACHINE\SOFTWARE\POLICIES\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET SETTINGS
0x8876e530   3028      0x138    0x20019 Key             USER\S-1-5-21-2833823845-3085568943-3082117713-1000\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET SETTINGS
0x95afbb28   3028      0x13c    0x20019 Key             MACHINE\SOFTWARE\POLICIES
0x8da88c20   3028      0x140    0x20019 Key             USER\S-1-5-21-2833823845-3085568943-3082117713-1000\SOFTWARE\POLICIES
0x920e8d20   3028      0x144    0x20019 Key             USER\S-1-5-21-2833823845-3085568943-3082117713-1000\SOFTWARE
0x85a34240   3028      0x148    0x20019 Key             MACHINE\SOFTWARE
0x85a1f580   3028      0x14c    0x20019 Key             MACHINE\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET SETTINGS
0x8f295030   3028      0x184    0x20019 Key             MACHINE\SYSTEM\CONTROLSET001\SERVICES\WINSOCK2\PARAMETERS\PROTOCOL_CATALOG9
0x85f7a740   3028      0x18c    0x20019 Key             MACHINE\SYSTEM\CONTROLSET001\SERVICES\WINSOCK2\PARAMETERS\NAMESPACE_CATALOG5
0x937d7350   3028      0x19c    0x20019 Key             USER\S-1-5-21-2833823845-3085568943-3082117713-1000\SOFTWARE\MICROSOFT\INTERNET EXPLORER\MAIN
0x93604bb8   3028      0x1a0    0x20019 Key             MACHINE\SOFTWARE\MICROSOFT\INTERNET EXPLORER\MAIN
0x85f5a7a8   3028      0x1a4    0xf003f Key             USER\S-1-5-21-2833823845-3085568943-3082117713-1000_CLASSES
0x864d7730   3028      0x1b4    0x20019 Key             USER
0x9212d3b0   3028      0x1b8        0x3 Key             USER\S-1-5-21-2833823845-3085568943-3082117713-1000
0x87980dc8   3028      0x1d8    0x20019 Key             MACHINE\SOFTWARE\MICROSOFT\TRACING\JACKAL_RASAPI32
0x95bf8030   3028      0x1f0    0x20019 Key             MACHINE\SOFTWARE\MICROSOFT\TRACING\JACKAL_RASMANCS
0x85fc2300   3028      0x248    0x20019 Key             USER\S-1-5-21-2833823845-3085568943-3082117713-1000\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET SETTINGS\ZONEMAP
0x865206c0   3028      0x24c    0x20019 Key             USER\S-1-5-21-2833823845-3085568943-3082117713-1000\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\NETWORK\LOCATION AWARENESS
0x85a29580   3028      0x258        0x9 Key             MACHINE\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\IMAGE FILE EXECUTION OPTIONS
0x8da179f0   3028      0x25c        0x8 Key             MACHINE\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION\APPCOMPATFLAGS
0x936a4b30   3028      0x264        0x8 Key             USER\S-1-5-21-2833823845-3085568943-3082117713-1000\SOFTWARE\MICROSOFT\WINDOWS NT\CURRENTVERSION
0x95a3de80   3028      0x268    0x20019 Key             MACHINE\SOFTWARE\MICROSOFT\WINDOWS\CURRENTVERSION\INTERNET SETTINGS\ZONEMAP
0x864a2d20   3028      0x270    0x20019 Key             MACHINE\SOFTWARE\MICROSOFT\INTERNET EXPLORER\MAIN\FEATURECONTROL\FEATURE_LOCALMACHINE_LOCKDOWN
0x85f5e3b8   3028      0x290    0x20019 Key             MACHINE\SOFTWARE\MICROSOFT\INTERNET EXPLORER\MAIN\FEATURECONTROL\FEATURE_ZONE_ELEVATION
```

Figure 2: Active Registry Handles from Malicious Process

   The open handle to `\REGISTRY\USER\[ID]\Software\Microsoft\Windows Player` suggests active modification of this key for persistence, as it stores Base64-encoded C2 configurations

6. Specifically, what values or data the malware add to the registry key? Is it possible using only the memory dump to find out? To query a cached registry key, use the printkey plugin like this: `"volatility -f <FILE> printkey -K "Software\Microsoft\The\Key\To\Find""` (3 Marks)

   Answer:

   Using Volatility's printkey plugin, it is possible to retrieve this registry key and its values directly from memory, without requiring disk access.

   The command used was:

```
volatility -f jackal.vmem --profile=Win7SP1x86_23418 printkey -K "Software\Microsoft\
    Windows Player"
```

   The memory dump analysis reveals the malware modified the registry key `Software\Microsoft\WindowsPlayer`, storing multiple Base64-encoded string values. These likely contain C2 payloads, supporting earlier reports of the malware using Base64/XOR encoding. Output is presented in Figure 3.

```
vagrant@lab2:~$ volatility -f jackal.vmem --profile=Win7SP1x86_23418 printkey -K "Software\Microsoft\Windows Player"
Volatility Foundation Volatility Framework 2.6.1
Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: \??\C:\Users\Daniel\ntuser.dat
Key name: Windows Player (S)
Last updated: 2013-03-11 16:22:02 UTC+0000

Subkeys:

Values:
REG_SZ        DB1L            : (S) IionPSEjKj0rKj0nIjxgdmF6fGA8e3J/cXZhPXtnfn8=
REG_SZ        WN33            : (S) ISEqPSY9ISAgPSEjJDxndnpqajxgY39meD1yYGM=
REG_SZ        4H2N            : (S) IismPSEhKj0iJiQ9IiUrPHx9YXZwanZ7fDxldmFxcn89Z2tn
REG_SZ        MRRU            : (S) IiUrPSsnPSIqKz0hJys8YHZ2fTxqdn9jPXtnfn8=
REG_SZ        HNFY            : (S) Jic9IiAqPSIrIz0iICs8YXxyYWA8ZXJhenxmdzxyf2d2YX1yZ3p8fT17Z35/
REG_SZ        IEUH            : (S) ISYgPSEhJz0iJCI9ICo8fXZkYDxwfH5+PWN7Yw==
REG_SZ        1AUR            : (S) IisnPScrPSInID0iIiQ8ZHJ6Z3p9dD1yYGM=
REG_SZ        47SG            : (S) ISIhPScgPSInIz0iJiE8YGN8YWc8e3Jje3JpcmF3PXlg
REG_SZ        FAU1            : (S) ISYhPSEhKj0iKiA9ISEkPGBnfHB4PGdhcnd6fXQ9e2d+fw==
REG_SZ        2LHL            : (S) ISAgPSIkIT0iKyM9IiUgPGFyfXB7dmE8ZHp9d3xkYD1je2M=
REG_SZ        5WYY            : (S) IiclPSshPSQkPSYqPGAqIioiIjx4f2B5ciIiPHV6f2d2YT1na2c=
REG_SZ        KYKG            : (S) IiUlPSEjPSYgPSEiKjxRZmFxcn14PF9mcHp3PHlycHhgPXtnfn8=
REG_SZ        Q810            : (S) IiYqPSEgIT0iIyE9IiYrPGNyYGB6fXQ8YHxmYXB2PHt8fnY9cmBj
REG_SZ        M65P            : (S) ISMhPSIqPSIqJD0iICI8fHBnfHF2YTxgcmdmYX09eWA=
REG_SZ        0GF9            : (S) JSQ9KyY9IScrPSEmPGRyenVgPHlmfXw9e2d+fw==
REG_SZ        IYCD            : (S) IiErPSIgJz0iJCU9IiElPHZremBnYDxDcmByd3Z9cj13fHA=
REG_SZ        78OI            : (S) IiMrPSIiJj0qKj0rJTxqdn9/fGR7cn5+dmE8YXZjfGFnYA==
REG_SZ        YBTI            : (S) ICE9ISc9IScrPSIkKzx0fHx3emB7PHV2f398ZD17Z35/
REG_SZ        THRG            : (S) IiMmPSQiPSEgJD0iJTxgZmN2YTxwf3JxcXZhYD1rfn8=
REG_SZ        PXDT            : (S) KiM9ICY9ISAnPSEnKzx8YXp2fWc8cXJ4dmFgPXtnfn8=
REG_SZ        FYNO            : (S) IiA9KyE9IiYiPSEiJjx/fGNjdmE8d2Z+cXF2f389Z2tn
REG_SZ        IWT5            : (S) IiMiPSogPSohPSIlJDx8YGd2fWB8YWo8ZHpweHZ3PHJ/fw==
REG_SZ        6NLE            : (S) IiQrPSsmPSIqIj0mITx2dmNhfH48YHZldmF2PXtnfn8=
REG_SZ        XOJV            : (S) ISIrPSIiID0iJCs9JCI8ZHZyZ3t2YT17Z35/
REG_SZ        XP8X            : (S) ISEkPSEhPSImJD0mPGZgcjxgfHBwdmE9e2d+fw==
REG_SZ        3EDQ            : (S) JyY9IiYiPSIrID0iJyo8eHJge3ZhPEt2YXxren10PXlg
REG_SZ        ONON            : (S) ISEiPSIlJj0iJSc9JiU8Y2Z/f3FycHg8anZge3plcns9Z2tn
REG_SZ        ZXU6            : (S) ISMjPSInJz0mIz0hIiE8YXZgcn5jf3Z3PHF2dXxhdj17Z35/
REG_SZ        A3D7            : (S) ISElPSEjIj0iJCA9JCE8e3x8d2R6fXg8RXJ9cHxmZXZhPGV6YHpnPGBmfn52YT17Z35/
REG_SZ        FRRM            : (S) JyM9IiM9ISAqPSEiKzx5emBnPXlg
```

Figure 3: Registry dump showing suspicious values under `Windows Player` key

After analysing the complete memory dump, it is not possible to fully interpret the values or data stored in the registry key alone, as they appear to be encrypted and require additional information for thorough analysis. The most that can be done at this stage is to decode the values, as I have demonstrated using `CyberChef` as shown in in Table 1. The Keys eg. `DB1L, WN33, 4H2N, etc...` were not possible to be decoded using the provided information.

Table 1: Registry dump showing Base64-decoded values under `Windows Player` key.

Key Name	IP Address	Path
DB1L	194.209.89.41	/serios/halber.html
WN33	229.5.233.207	/teiyy/spluk.asp
4H2N	185.229.157.168	/onrecyeho/verbal.txt
MRRU	168.84.198.248	/seen/yelp.html
HNFY	54.139.180.138	/roars/varioud/alternation.html
IEUH	253.224.171.39	/news/comm.php
1AUR	184.48.143.117	/waiting.asp
47SG	212.43.140.152	/sport/haphazard.js
FAU1	252.229.193.227	/stock/trading.html
2LHL	233.172.180.163	/rancher/windows.php
5WYY	146.82.77.59	/s91911/klsja11/filter.txt
KYKG	166.20.53.219	/Burbank/Lucid/jacks.html
Q810	159.232.102.158	/passing/source/home.asp
M65P	202.19.197.131	/october/saturn.js
0GF9	67.85.248.25	/waifs/juno.html
IYCD	128.134.176.126	/exists/Pasadena.doc
78OI	108.115.99.86	/yellowhammer/reports
YBTI	32.24.248.178	/goodish/fellow.html
THRG	105.71.237.16	/super/clabbers.xml
PXDT	90.35.234.248	/orient/bakers.html
FYNO	13.82.151.215	/lopper/dumbbell.txt
IWT5	101.93.92.167	/ostensory/wicked/all
6NLE	178.85.191.52	/eeprom/severe.html
XOJV	218.113.178.71	/weather.html
XP8X	227.22.157.5	/usa/soccer.html
3EDQ	45.151.183.149	/kasher/Xeroxing.js
ONON	221.165.164.56	/pullback/yeshivah.txt
ZXU6	200.144.50.212	/resampled/before.html
A3D7	226.201.173.72	/hoodwink/Vancouver/visit/summer.html
FRRM	40.10.239.218	/jist.js

Each value contained a Base64-encoded string, which likely stores configuration data, such as **Command-and-control (C2) server URLs, File paths or script names** and **payload delivery**

7. Are there any specific network artefacts that you would configure an Intrusion Detection System to look for? (2 Marks)

   Answer:
   Yes, configure the IDS to detect:

   C2 Communications:

   - HTTP requests with the unique `User-Agent` string: `"The Jackal v4.2001"`
   - Base64 strings in network traffic matching the registry patterns

   Suspicious Activity:

   - Unusual outbound `POST` requests to unknown IPs or ports
   - Connections originating from or targeting the malware path: `c:\windows\system32\jackal.exe`

8. What's the name of the mutex this malware uses? (3 Marks)

   Answer:
   The malware uses the mutex `__Dassara__`, identified via Volatility's `mutantscan` output tied to PID `3028` as shown in Figure 4. This non-standard name confirms malicious use for process synchronisation.

   Evidence:

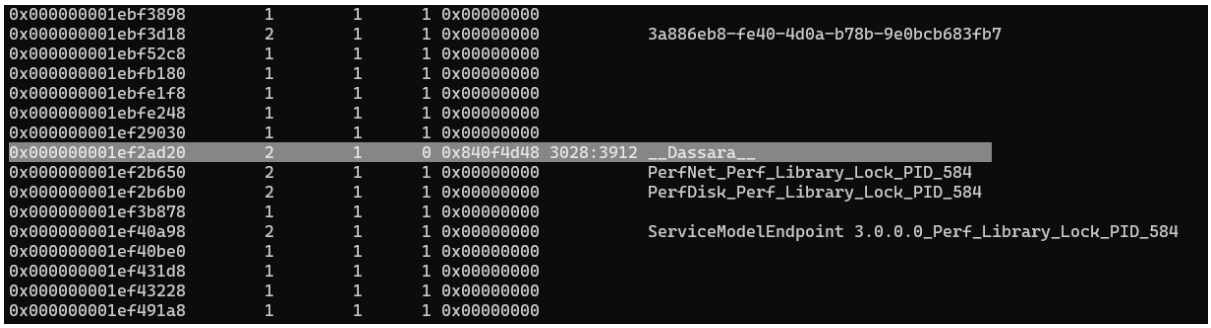

Figure 4: Volatility mutantscan output

9. List as many IPs/hostnames used by the malware. Is the guy right about it using base64 and xor? (3 Marks)

   Answer:
   IPs/Hostnames Used by the Malware:

   Primary C2 Server:

   - `172.16.237.1` (Foreign Address)
   - Connected to malware's listener on `172.16.237.134:9090` (State: `ESTABLISHED/CLOSE_WAIT`)
   - Likely the attacker's machine or part of the C2 infrastructure

   Malware-Controlled Host:

   - `172.16.237.134` (Local Address)
   - Actively listening on port `9090` (PID `3028`: `jackal.exe.exe`)

   IPv6 Address:

   - `fe80::bdff:e616:7b50:1e75`

   25+ decrypted IP address have been displayed in Table 1. It can be referenced to get further list of IPs/hostnames and paths used by the Malware. Also a screenshot has been presented in Figure 5 to demonstrate the way the IPs were decrypted.

Base64/XOR Validation:

Yes, the guy was correct, Jackal's C2 list is just base64 and xor:

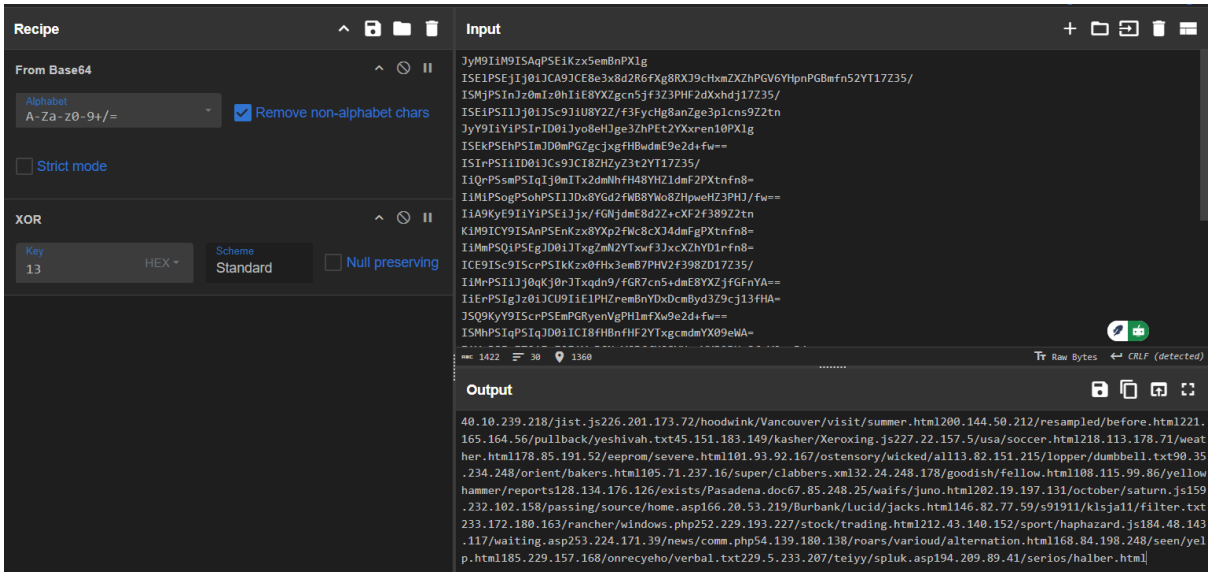

Figure 5: Decoded registry values reveal a list of suspicious URLs and file paths and IP adderesses

10. Can you detect network activity from the malware and if so which network protocol is it using, on which local port and what is it doing (what is the state)? (4 Marks)

Answer:

A screenshot of the command `volatility -f jackal.vmem --profile=Win7SP1x86_23418 netscan` is presented in Figure 6. Which was used to get the readings presented below.

The malware uses multiple network protocols:

- **TCPv4/TCPv6:** For command-and-control (C2) communications
- **UDPv4/UDPv6:** For scanning and broadcast purposes

Key Local Ports and States:

Primary Malware Port:

- `TCP 9090` (`LISTENING`) via PID `3028` (`jackal.exe.exe`)
- Connections to `172.16.237.1` in `ESTABLISHED` and `CLOSE_WAIT` states

Other Suspicious Ports:

- `TCP 135, 445, 49152--49156` (potential lateral movement)
- `UDP 137, UDP 138` (NetBIOS — possible network scanning)
- `UDP 5355` (mDNS — service discovery)

State Analysis:

- `LISTENING (TCP 9090):` Persistent backdoor
- `ESTABLISHED (TCP 9090 ↔ 172.16.237.1):` Active C2 session
- `CLOSE_WAIT:` Recent session termination
- `UDP` ports show broadcast/listening behavior

Operational Purpose:

- `TCP 9090` is the primary C2 communication channel

- Other ports suggest:
  - Network reconnaissance (`UDP`)
  - Secondary persistence mechanisms (high-numbered `TCP` ports)
  - Possible lateral movement (`SMB/NetBIOS` ports)

```
vagrant@lab2:~$ volatility -f jackal.vmem --profile=Win7SP1x86_23418 netscan
Volatility Foundation Volatility Framework 2.6.1
Offset(P)          Proto    Local Address                  Foreign Address      State            Pid      Owner          Created
0x1e93c570         UDPv4    0.0.0.0:5355                   *:*                                   1064     svchost.exe    2013-03-11 16:22:02 UTC+0000
0x1e93c570         UDPv6    :::5355                        *:*                                   1064     svchost.exe    2013-03-11 16:22:02 UTC+0000
0x1e802838         TCPv4    0.0.0.0:49156                  0.0.0.0:0            LISTENING        480      services.exe
0x1e826500         TCPv4    172.16.237.134:9090            172.16.237.1:50645   ESTABLISHED      -1
0x1e92b008         TCPv4    172.16.237.134:9090            172.16.237.1:50606   CLOSE_WAIT       -1
0x1ef6b740         UDPv4    172.16.237.134:138             *:*                                   4        System         2013-03-11 15:00:22 UTC+0000
0x1ef6d450         UDPv4    172.16.237.134:137             *:*                                   4        System         2013-03-11 15:00:22 UTC+0000
0x1ea34940         TCPv4    172.16.237.134:139             0.0.0.0:0            LISTENING        4        System
0x1eac1008         TCPv4    0.0.0.0:135                    0.0.0.0:0            LISTENING        668      svchost.exe
0x1eac1008         TCPv6    :::135                         :::0                 LISTENING        668      svchost.exe
0x1eac2ca8         TCPv4    0.0.0.0:135                    0.0.0.0:0            LISTENING        668      svchost.exe
0x1eaf60d8         TCPv4    0.0.0.0:49155                  0.0.0.0:0            LISTENING        820      svchost.exe
0x1eb338c8         TCPv4    0.0.0.0:9090                   0.0.0.0:0            LISTENING        3028     jackal.exe.exe
0x1ebfa140         TCPv4    0.0.0.0:445                    0.0.0.0:0            LISTENING        4        System
0x1ebfa140         TCPv6    :::445                         :::0                 LISTENING        4        System
0x1ebfc490         TCPv4    0.0.0.0:49156                  0.0.0.0:0            LISTENING        480      services.exe
0x1ebfc490         TCPv6    :::49156                       :::0                 LISTENING        480      services.exe
0x1ef4b370         TCPv4    0.0.0.0:49153                  0.0.0.0:0            LISTENING        756      svchost.exe
0x1ef4b370         TCPv6    :::49153                       :::0                 LISTENING        756      svchost.exe
0x1ef569f0         TCPv4    0.0.0.0:49154                  0.0.0.0:0            LISTENING        488      lsass.exe
0x1ef569f0         TCPv6    :::49154                       :::0                 LISTENING        488      lsass.exe
0x1ef64c78         TCPv4    0.0.0.0:49154                  0.0.0.0:0            LISTENING        488      lsass.exe
0x1fa91520         UDPv4    0.0.0.0:0                      *:*                                   1064     svchost.exe    2013-03-11 16:22:02 UTC+0000
0x1fa91520         UDPv6    :::0                           *:*                                   1064     svchost.exe    2013-03-11 16:22:02 UTC+0000
0x1faa1280         UDPv4    0.0.0.0:5355                   *:*                                   1064     svchost.exe    2013-03-11 16:22:02 UTC+0000
0x1faef990         UDPv6    fe80::bdff:e616:7b50:1e75:546  *:*                                   756      svchost.exe    2013-03-11 16:33:11 UTC+0000
0x1f7c3378         TCPv4    0.0.0.0:49153                  0.0.0.0:0            LISTENING        756      svchost.exe
0x1f7c9150         TCPv4    0.0.0.0:49152                  0.0.0.0:0            LISTENING        384      wininit.exe
0x1f7c9150         TCPv6    :::49152                       :::0                 LISTENING        384      wininit.exe
0x1f7ca2a8         TCPv4    0.0.0.0:49152                  0.0.0.0:0            LISTENING        384      wininit.exe
0x1fa09f60         TCPv4    0.0.0.0:49155                  0.0.0.0:0            LISTENING        820      svchost.exe
0x1fa09f60         TCPv6    :::49155                       :::0                 LISTENING        820      svchost.exe
0x1fb17b00         TCPv6    fe80::bdff:e616:7b50:1e75:49168 ff02::1:2:445       CLOSED           -1
0x1fd8f008         TCPv6    fe80::bdff:e616:7b50:1e75:445  ff02::1:2:49168      CLOSED           -1
```

Figure 6: Volatility `netscan` output showing active network connections from the memory dump.