BankingTroubles

Company X has contacted you to perform forensics work on a recent incident that occurred. One of their employees had received an e-mail from a co-worker that pointed to a PDF file. Upon opening, the employee did not notice anything; however, they recently had unusual activity in their bank account.

The initial theory is that a user received an e-mail, containing an URL leading to a forged PDF document. Opening that document in Acrobat Reader triggers a malicious Javascript that initiates a sequence of actions to take over the victim's system.

Company X was able to obtain a memory image of the employee's virtual machine upon suspected infection and asked you as a security blue team analyst to analyze the virtual memory and provide answers to the questions.

Task 1:
What was the local IP address of the victim's machine?

I used the connscan plugin and found the local IP

```
remnux@remnux:~/volatility$ python2 vol.py -f '/home/remnux/Bob.vmem' --profile=WinXPSP2x86 connscan
Volatility Foundation Volatility Framework 2.6.1
Offset(P)  Local Address             Remote Address            Pid
---------- ------------------------ ------------------------ ---
0x01e6a9f0 192.168.0.176:1176       212.150.164.203:80        888
0x01ec57c0 192.168.0.176:1189       192.168.0.1:9393          1244
0x01ed4270 192.168.0.176:2869       192.168.0.1:30379         1244
0x01eef808 192.168.0.176:2869       192.168.0.1:30380         4
0x01ffa7f8 0.0.0.0:0                80.206.204.129:0          0
0x02041108 127.0.0.1:1168           127.0.0.1:1169            888
0x0225a448 192.168.0.176:1172       66.249.91.104:80          888
0x0226ac58 127.0.0.1:1169           127.0.0.1:1168            888
0x0227ac58 192.168.0.176:1171       66.249.90.104:80          888
0x02308890 192.168.0.176:1178       212.150.164.203:80        1752
0x02323008 192.168.0.176:1184       193.104.22.71:80          880
0x02410440 192.168.0.176:1185       193.104.22.71:80          880
```

Answer: 192.168.0.176

Task 2:
What was the OS environment variable's value?

I used the verinfo plugin and found the OS

```
C:\WINDOWS\system32\RPCRT4.dll
  File version    : 5.1.2600.2180
  Product version : 5.1.2600.2180
  Flags           :
  OS              : Windows NT
  File Type       : Dynamic Link Library
  File Date       :
```

Answer: Windows NT

Task 3:
What was the Administrator's password?

I used the hashdump plugin and then cracked it inside hashes.com

```
remnux@remnux:~/volatility$ python2 vol.py -f '/home/remnux/Bob.vmem' --profile=WinXPSP2x86 hashdump
Volatility Foundation Volatility Framework 2.6.1
Administrator:500:e52cac67419a9a224a3b108f3fa6cb6d:8846f7eaee8fb117ad06bdd830b7586c:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
HelpAssistant:1000:9f8ac2eaebcd2e3a6f94d53c19803662:d95e38a172b3ddaa1ce0b63bb1f5e1fb:::
SUPPORT_388945a0:1002:aad3b435b51404eeaad3b435b51404ee:ad052c1cbab3ec2502df165cd25d95bd:::
remnux@remnux:~/volatility$
```

✔ Found:

```
8846f7eaee8fb117ad06bdd830b7586c:password
```

Answer: password

Task 4:
Which process was most likely responsible for the initial exploit?

I used the pstree plugin and noticed that the firefox.exe was the parent process of AcroRd32.exe which is suspicious

```
remnux@remnux:~/volatility$ python2 vol.py -f '/home/remnux/Bob.vmem' --profile=WinXPSP2x86 pstree
Volatility Foundation Volatility Framework 2.6.1
Name                                                  Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ ----
 0x81cdd790:explorer.exe                             1756   1660     14    345 2010-02-26 03:34:38 UTC+0000
. 0x820cd5c8:VMwareUser.exe                          1116   1756      4    179 2010-02-26 03:34:39 UTC+0000
. 0x81ca96f0:VMwareTray.exe                          1108   1756      1     59 2010-02-26 03:34:39 UTC+0000
. 0x82068020:firefox.exe                              888   1756      9    172 2010-02-27 20:11:53 UTC+0000
.. 0x820618c8:AcroRd32.exe                           1752    888      8    184 2010-02-27 20:12:23 UTC+0000
```

Answer: AcroRd32.exe

Task 5:
What is the extension of the malicious file retrieved from the process responsible for the initial exploit?

If the initial exploit is AcroRd32.exe this means the file is pdf extension.

Answer: pdf

Task 6:
Suspicious processes opened network connections to external IPs. One of them starts with "2". Provide the full IP.

I used the plugin connscan and found the IP

```
remnux@remnux:~/volatility$ python2 vol.py -f '/home/remnux/Bob.vmem' --profile=WinXPSP2x86 connscan
Volatility Foundation Volatility Framework 2.6.1
Offset(P)  Local Address             Remote Address            Pid
---------- ------------------------ ------------------------ ---
0x01e6a9f0 192.168.0.176:1176       212.150.164.203:80        888
0x01ec57c0 192.168.0.176:1189       192.168.0.1:9393          1244
0x01ed4270 192.168.0.176:2869       192.168.0.1:30379         1244
0x01eef808 192.168.0.176:2869       192.168.0.1:30380         4
0x01ffa7f8 0.0.0.0:0                80.206.204.129:0          0
0x02041108 127.0.0.1:1168           127.0.0.1:1169            888
0x0225a448 192.168.0.176:1172       66.249.91.104:80          888
0x0226ac58 127.0.0.1:1169           127.0.0.1:1168            888
0x0227ac58 192.168.0.176:1171       66.249.90.104:80          888
0x02308890 192.168.0.176:1178       212.150.164.203:80        1752
0x02323008 192.168.0.176:1184       193.104.22.71:80          880
0x02410440 192.168.0.176:1185       193.104.22.71:80          880
```

Answer: 212.150.164.203

Task 7:
A suspicious URL was present in process svchost.exe memory. Provide the full URL that points to a PHP page hosted over a public IP (no FQDN).

In task 6, the PID of the network connection is 880 and inside pstree the PID 880 is associate with svchost.exe

```
... 0x82266870:svchost.exe                           880    688     28    340 2010-02-26 03:34:07 UTC+0000
```

I used memdump to dump the memory of the svchost process and used strings on it with grep on php

```
remnux@remnux:~/volatility$ python2 vol.py -f '/home/remnux/Bob.vmem' --profile=WinXPSP2x86 memdump --pid=880 --dump-dir=/home/remnux/svchost
Volatility Foundation Volatility Framework 2.6.1
************************************************************************
Writing svchost.exe [   880] to 880.dmp
```

```
remnux@remnux:~/svchost$ strings 880.dmp | grep -i php
http://193.104.22.71/~produkt/9j856f_4m9y8urb.php
POST /~produkt/9j856f_4m9y8urb.php HTTP/1.1
POST /~produkt/9j856f_4m9y8urb.php HTTP/1.1
http://193.104.22.71/~produkt/9j856f_4m9y8urb.php
http://193.104.22.71/~produkt/9j856f_4m9y8urb.php
http://193.104.22.71/~produkt/9j856f_4m9y8urb.php
http://193.104.22.71/~produkt/9j856f_4m9y8urb.php
b.php
http://193.104.22.71/~produkt/9j856f_4m9y8urb.php
b.php
/~produkt/9j856f_4m9y8urb.php
/~produkt/9j856f_4m9y8urb.php
http://193.104.22.71/~produkt/9j856f_4m9y8urb.php&N
php?X
http://193.104.22.71/~produkt/9j856f_4m9y8urb.php
http://193.104.22.71/~produkt/9j856f_4m9y8urb.php
```

Answer: http://193.104.22.71/~produkt/9j856f_4m9y8urb.php

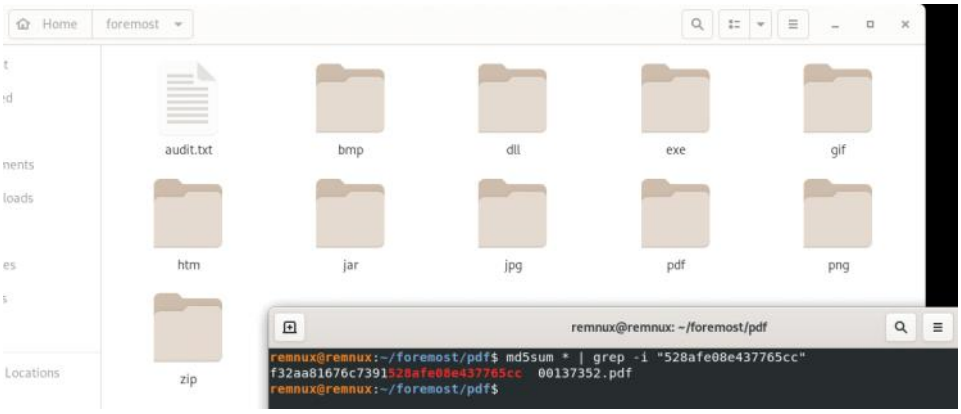
Task 8:
Extract files from the initial process. One file has an MD5 hash ending with "528afe08e437765cc". When was this file first submitted for analysis on VirusTotal?

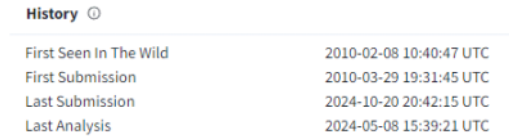
I used the foremost tool to extract files from the memory dump in task 7.

The command I used is

foremost -i /home/remnux/svchost/880.dmp -o '/home/remnux/foremost'

Then I used grep on the MD5 to find the full hash





Answer: 2010-03-29 19:31:45

Task 9:
What was the PID of the process that loaded the file PDF.php?

In the pstree plugin we can the the AcroRd32.exe PID is 1752

```
remnux@remnux:~/volatility$ python2 vol.py -f '/home/remnux/Bob.vmem' --profile=WinXPSP2x86 pstree
Volatility Foundation Volatility Framework 2.6.1
Name                                                  Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ ----
 0x81cdd790:explorer.exe                             1756   1660     14    345 2010-02-26 03:34:38 UTC+0000
. 0x820cd5c8:VMwareUser.exe                          1116   1756      4    179 2010-02-26 03:34:39 UTC+0000
. 0x81ca96f0:VMwareTray.exe                          1108   1756      1     59 2010-02-26 03:34:39 UTC+0000
. 0x82068020:firefox.exe                              888   1756      9    172 2010-02-27 20:11:53 UTC+0000
.. 0x820618c8:AcroRd32.exe                           1752    888      8    184 2010-02-27 20:12:23 UTC+0000
```

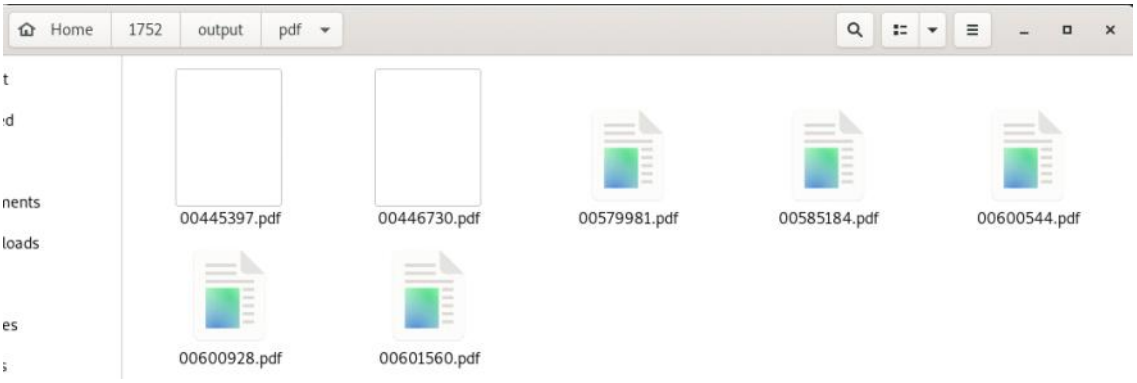
Answer: 1752

Task 10:
The JS includes a function meant to hide the call to function eval(). Provide the name of that function.

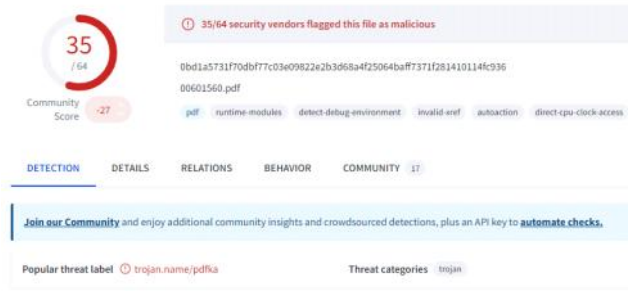
I dumped the memory of the PID 1752

```
remnux@remnux:~/volatility$ python2 vol.py -f '/home/remnux/Bob.vmem' --profile=WinXPSP2x86 memdump -p 1752 -D '/home/remnux/1752'
Volatility Foundation Volatility Framework 2.6.1
************************************************************************
Writing AcroRd32.exe [  1752] to 1752.dmp
```

Then I used foremost on the dmp file and navigated to the pdf folder



Then I checked the MD5 of the pdf files in Virus Total and the only malicious pdf was 00601560.pdf - f32aa81676c7391528afe08e437765cc



Then I used peepdf tool to identify if the javascript is embedded in this file

```
remnux@remnux:~/1752/output/pdf$ peepdf -f 00601560.pdf
Warning: PyV8 is not installed!!

File: 00601560.pdf
MD5: f32aa81676c7391528afe08e437765cc
SHA1: 6045554853a61681d7264260cdd1072bbdc113ac
SHA256: 0bd1a5731f70dbf77c03e09822e2b3d68a4f25064baff7371f281410114fc936
Size: 607083 bytes
Version: 1.3
Binary: False
Linearized: False
Encrypted: False
Updates: 0
Objects: 5
Streams: 1
URIs: 0
Comments: 0
Errors: 1

Version 0:
        Catalog: 112
        Info: No
        Objects (5): [11, 112, 787, 1054, 1847]
        Streams (1): [1054]
                Encoded (1): [1054]
        Objects with JS code (1): [1054]
        Suspicious elements:
                /AA (1): [1847]
                /JS (1): [11]
                /JavaScript (1): [11]
```

I asked the Chat about the output

It looks like the JavaScript code is embedded in **object 1054** of the PDF. The next step is to extract this JavaScript code using `pdf-parser.py` and then beautify it for analysis.

The **1054** in this context refers to the **object number** in the PDF file that contains the embedded JavaScript code.

When analyzing a PDF file, especially one that might be malicious, tools like **peepdf** will list the different objects within the PDF. Each object can have various types of content, such as text, images, or embedded JavaScript. The object numbers help you identify and extract specific elements of the PDF.

In your output from peepdf, it showed:

less — Copy code

```
Objects with JS code (1): [1054]
Streams (1): [1054]
```

This means that object 1054 is the one containing JavaScript code embedded as a stream in the PDF. By using **1054** as the object number with tools like `pdf-parser.py`, you can extract the content of this object, which in this case, is likely to be the obfuscated JavaScript.

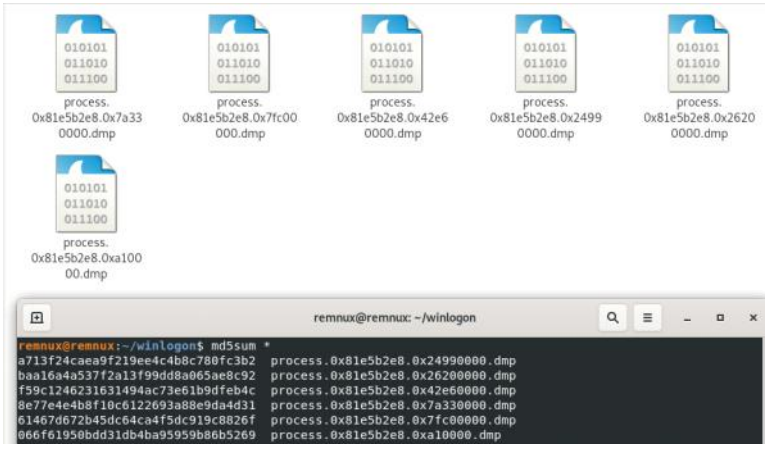
Task 12:
Process winlogon.exe hosted a popular malware that was first submitted for analysis at VirusTotal on 2010-03-29 11:34:01. Provide the MD5 hash of that malware.

I used the malfind plugin and grepped for winlogon

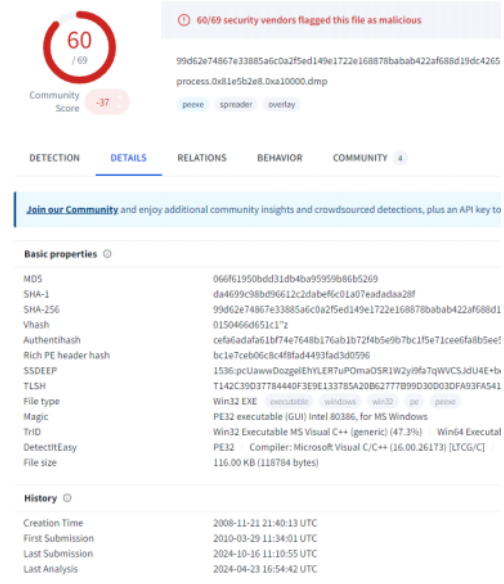
```
remnux@remnux:~/volatility$ python2 vol.py -f '/home/remnux/Bob.vmem' --profile=WinXPSP2x86 malfind | grep -i winlogon
Volatility Foundation Volatility Framework 2.6.1
Process: winlogon.exe Pid: 644 Address: 0xa10000
Process: winlogon.exe Pid: 644 Address: 0x24990000
Process: winlogon.exe Pid: 644 Address: 0x42e60000
Process: winlogon.exe Pid: 644 Address: 0x26200000
Process: winlogon.exe Pid: 644 Address: 0x7a330000
Process: winlogon.exe Pid: 644 Address: 0x7fc00000
```

Then I saw only the PID 644 so I dumped it with the command

python2 vol.py -f '/home/remnux/Bob.vmem' --profile=WinXPSP2x86 malfind --pid=644 --dump-dir='/home/remnux/winlogon'



Only because the last MD5 was the shorter one so I checked him first in Virus Total and this one was the malicious



Answer: 066f61950bdd31db4ba95959b86b5269

Task 13:
What is the name of the malicious executable referenced in registry hive '\WINDOWS\system32\config\software', and is variant of ZeuS trojan?

I used the hivelist plugin to check for the hives and their offsets

```
remnux@remnux:~/volatility$ python2 vol.py -f '/home/remnux/Bob.vmem' --profile=WinXPSP2x86 hivelist
Volatility Foundation Volatility Framework 2.6.1
Virtual    Physical   Name
---------- ---------- ----
0xe1d6cb60 0x0abf7b60 \Device\HarddiskVolume1\Documents and Settings\Administrator\Local Settings\Application Data\Microsoft\Windows\UsrClass.dat
0xe1de0b60 0x0b68ab60 \Device\HarddiskVolume1\Documents and Settings\Administrator\NTUSER.DAT
0xe1769b60 0x069e2b60 \Device\HarddiskVolume1\Documents and Settings\LocalService\Local Settings\Application Data\Microsoft\Windows\UsrClass.dat
0xe17deb60 0x073f8b60 \Device\HarddiskVolume1\Documents and Settings\LocalService\NTUSER.DAT
0xe1797b60 0x06d3bb60 \Device\HarddiskVolume1\Documents and Settings\NetworkService\Local Settings\Application Data\Microsoft\Windows\UsrClass.dat
0xe17a3820 0x0e99c820 \Device\HarddiskVolume1\Documents and Settings\NetworkService\NTUSER.DAT
0xe1526748 0x036bd748 \Device\HarddiskVolume1\WINDOWS\system32\config\software
0xe15a3950 0x04351950 \Device\HarddiskVolume1\WINDOWS\system32\config\default
0xe151ea08 0x034b5a08 \Device\HarddiskVolume1\WINDOWS\system32\config\SAM
0xe153e518 0x03858518 \Device\HarddiskVolume1\WINDOWS\system32\config\SECURITY
0xe139d008 0x02e48008 [no name]
0xe1035b60 0x02a9db60 \Device\HarddiskVolume1\WINDOWS\system32\config\system
0xe102e008 0x02a97008 [no name]
```

Then I dumped the registry hive with the dumpregistry plugin

python2 vol.py -f '/home/remnux/Bob.vmem' --profile=WinXPSP2x86 dumpregistry -o 0xe1526748 -D '/home/remnux/software'

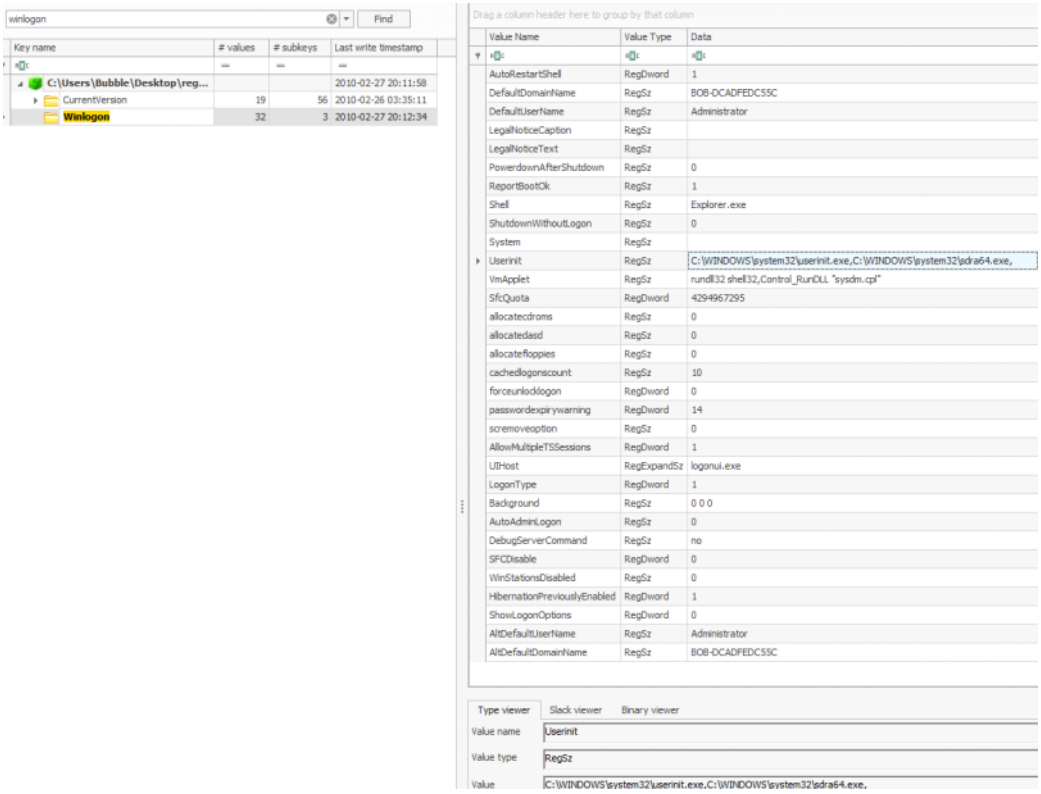
```
remnux@remnux:~/volatility$ python2 vol.py -f '/home/remnux/Bob.vmem' --profile=WinXPSP2x86 dumpregistry -o 0xe1526748 -D '/home/remnux/software'
Volatility Foundation Volatility Framework 2.6.1
**************************************************
Writing out registry: registry.0xe1526748.software.reg

**************************************************
```

Then I opened the hive with Registry Explorer but before I started to investigate this hive I first checked on Google for "Zeus trojan software hive" and found an article about the ZeuS Inforstealer Trojan
https://blogs.blackberry.com/en/2020/04/threat-spotlight-zeus-infostealer-trojan

Inside the article I searched for "Software" and I found the specific key which ZeuS modifies

ZeuS modifies the *userinit* subkey of the *HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Winlogon* registry entry to maintain persistence. It adds the path to *sdra64.exe* to the existing value of *C:\WINDOW\system32\userinit.exe*.

I checked the Winlogon inside the Registry Explorer and I found the same executable file like in the article picture



Answer: sdra64.exe

Task 10, 11,14,15 I did everything with the writeups after several failed attempts with the same tools in the hints and in the writeups, all tools didn't worked or gave other outputs from the writeups.