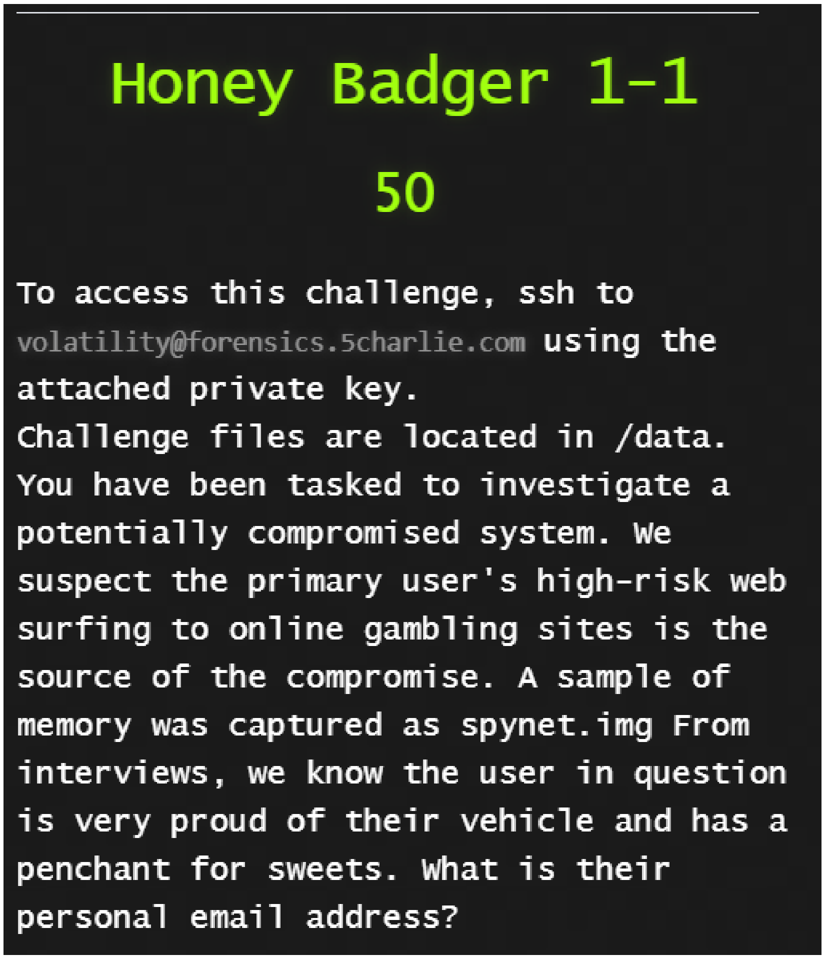


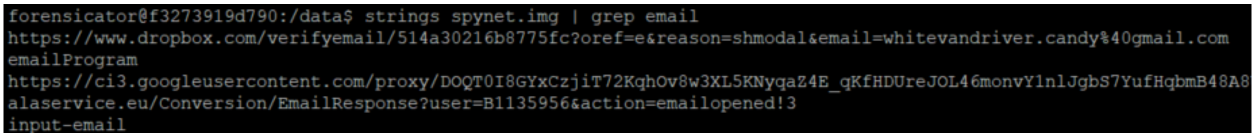
# Honey Badger 1-1

50

To access this challenge, ssh to volatility@forensics.5charlie.com using the attached private key.
Challenge files are located in /data.
You have been tasked to investigate a potentially compromised system. We suspect the primary user's high-risk web surfing to online gambling sites is the source of the compromise. A sample of memory was captured as spynet.img From interviews, we know the user in question is very proud of their vehicle and has a penchant for sweets. What is their personal email address?

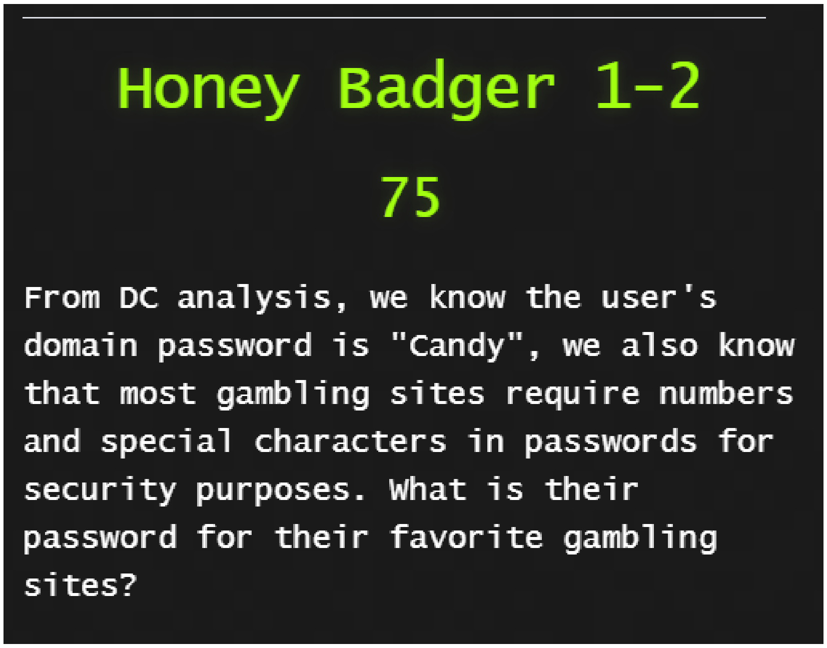
Profile: Win7SP1x86_24000

To start off we can run a strings on this image and grep for email to see if anything pops for us, and when we do it is clear that we see a line from dropbox that sent a verify email to our flag



```
forensicator@f3273919d790:/data$ strings spynet.img | grep email
https://www.dropbox.com/verifyemail/514a30216b8775fc?oref=e&reason=shmodal&email=whitevandriver.candy%40gmail.com
emailProgram
https://ci3.googleusercontent.com/proxy/DOQT0I8GYxCzjiT72KqhOv8w3XL5KNyqaZ4E_qKfHDUreJOL46monvY1nlJgbS7YufHqbmB48A8
alaservice.eu/Conversion/EmailResponse?user=B1135956&action=emailopened!3
input-email
```

Flag: whitevandriver.candy@gmail.com



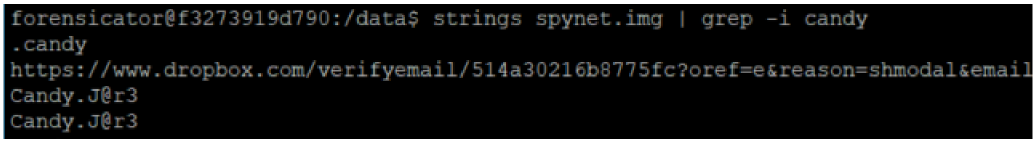
Honey Badger 1-2

75

From DC analysis, we know the user's domain password is "Candy", we also know that most gambling sites require numbers and special characters in passwords for security purposes. What is their password for their favorite gambling sites?

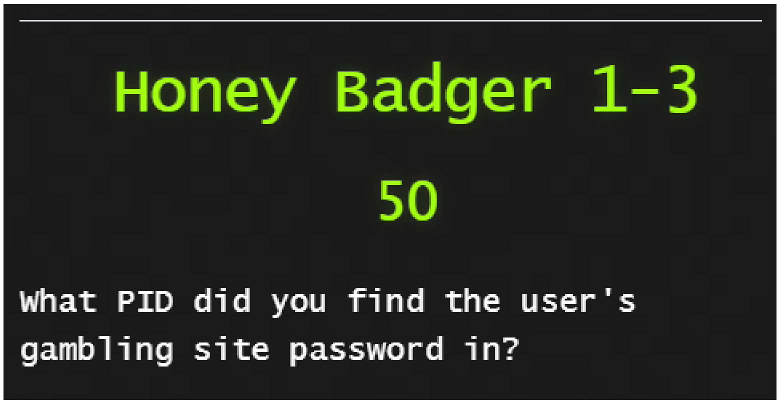
Again, we can try to grep for Candy to see if we have any luck finding a password reuse.

We are in luck again as the line right after dropbox shows a potential password.



```
forensicator@f3273919d790:/data$ strings spynet.img | grep -i candy
.candy
https://www.dropbox.com/verifyemail/514a30216b8775fc?oref=e&reason=shmodal&email
Candy.J@r3
Candy.J@r3
```

Flag: Candy.J@r3



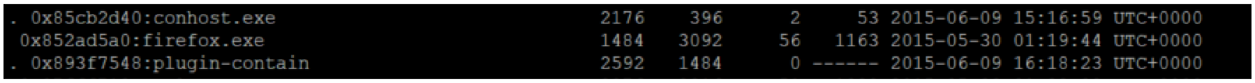
Honey Badger 1-3

50

What PID did you find the user's gambling site password in?

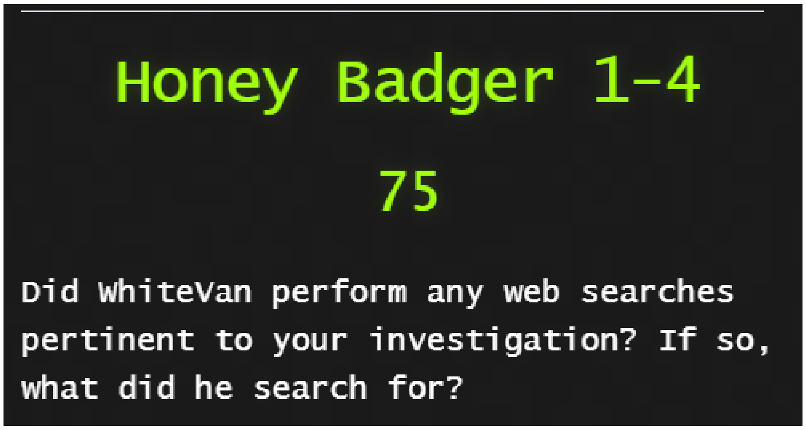
Going into volatility we see that the only web browser open on the system is firefox.exe at pid 1484

The previous 2 questions could probably have been found by dumping the process and strings the dump files.



```
. 0x85cb2d40:conhost.exe                            2176    396      2     53 2015-06-09 15:16:59 UTC+0000
 0x852ad5a0:firefox.exe                             1484   3092     56   1163 2015-05-30 01:19:44 UTC+0000
. 0x893f7548:plugin-contain                         2592   1484      0 ------ 2015-06-09 16:18:23 UTC+0000
```

Flag: 1484

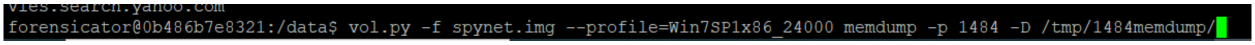


Honey Badger 1-4

75

Did WhiteVan perform any web searches pertinent to your investigation? If so, what did he search for?

For this we will need to memdump the Firefox process



```
vies.search.yahoo.com
forensicator@0b486b7e8321:/data$ vol.py -f spynet.img --profile=Win7SP1x86_24000 memdump -p 1484 -D /tmp/1484memdump/
```

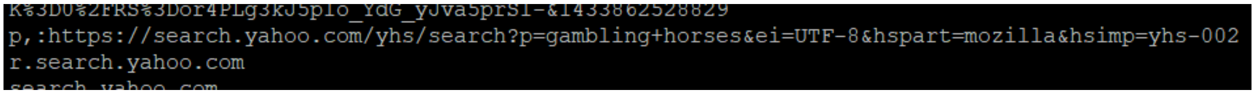
Once we have a dump of the memory, we can also run strings on that dump and save the output

Strings /tmp/1484memdump/1484.dmp > /tmp/1484memdump/strings

This next section can take some time, we can grep for the popular search engines until we find something interesting.

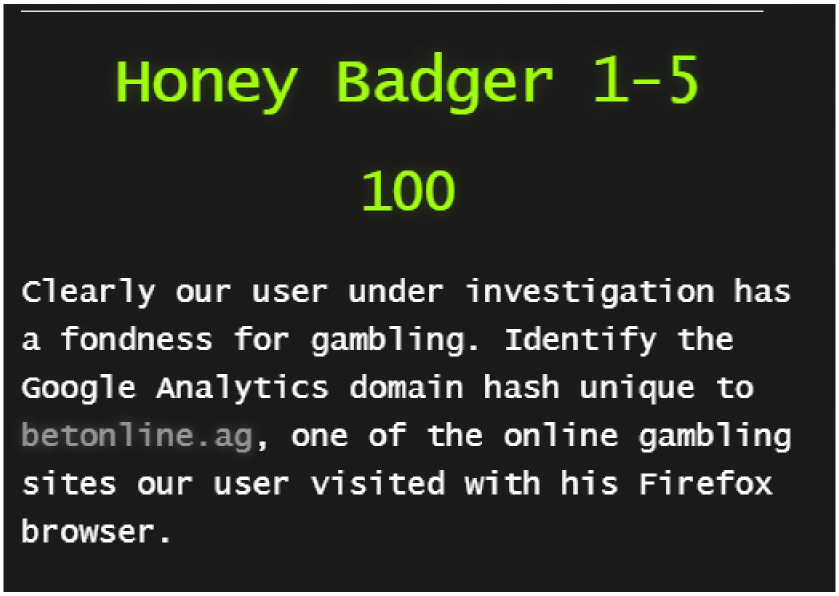
In this case google came back with lots of gibberish and after looking through had no results.

Grep search.yahoo.com we find some references to gambling online with an eventual line to a search that was completed.



```
K%3D0%2FRS%3Dor4PLg3kJ5plo_YdG_yJva5prSl-&1433862528829
p,:https://search.yahoo.com/yhs/search?p=gambling+horses&ei=UTF-8&hspart=mozilla&hsimp=yhs-002
r.search.yahoo.com
search.yahoo.com
```

Flag: gambling horses



Honey Badger 1-5

100

Clearly our user under investigation has a fondness for gambling. Identify the Google Analytics domain hash unique to betonline.ag, one of the online gambling sites our user visited with his Firefox browser.

I ran this command on the strings file: cat /tmp/1484memdump/strings | grep -i "google" | grep -i "analytics" | grep -i "betonline.ag"

And I came across this:

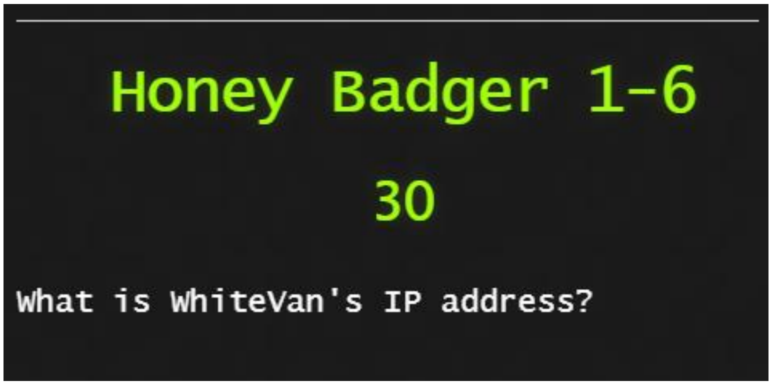
```
:http://www.google-analytics.com/r/__utm.gif?utmwv=5.6.4&utms=1&utmn=870245097&utmhn=www.betonline.ag&utmcs=UTF-8&utmsr=1280x800&utmvp=1280x658&utmsc=24-bit&utmul=en-us&utmje=0&utmfl=-&utmdt=You%20have%20successfully%20logged%20out!&utmhid=1271536205&utmr=-&utmp=%2Fcome-back-soon&utmht=1433865058159&utmac=UA-30537011-1&utmcc=__utma%3D203177346.934865508.1433862770.1433862770.1433865058.2%3B%2B__utmz%3D203177346.1433862770.1.1.utmcsr%3D(direct)%7Cutmccn%3D(direct)%7Cutmcmd%3D(none)%3B&utmjid=2071448188&utmredir=1&utmu=qAAAAAAAAAAAAAAAAAAAAAAE~
```

This is the important part we are looking for out of the string above:

203177346.934865508.1433862770.1433862770.1433865058.2

Where the first section to the. is the domain hash

Flag: 203177346

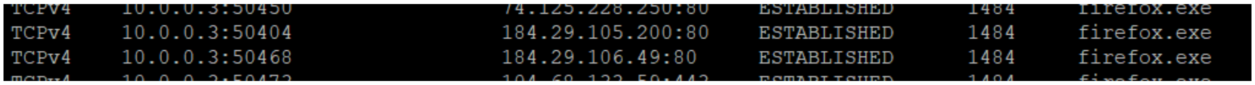


## Honey Badger 1-6

30

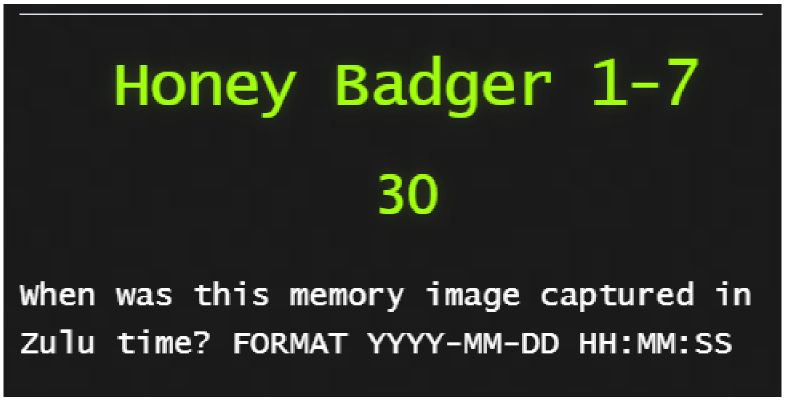
What is WhiteVan's IP address?

Run the netscan plugin to get the ip address.



```
TCPv4     10.0.0.3:50450                   74.125.228.250:80    ESTABLISHED       1484      firefox.exe
TCPv4     10.0.0.3:50404                   184.29.105.200:80    ESTABLISHED       1484      firefox.exe
TCPv4     10.0.0.3:50468                   184.29.106.49:80     ESTABLISHED       1484      firefox.exe
TCPv4     10.0.0.3:50472                   104.68.122.50:443    ESTABLISHED       1484      firefox.exe
```

Flag: 10.0.0.3

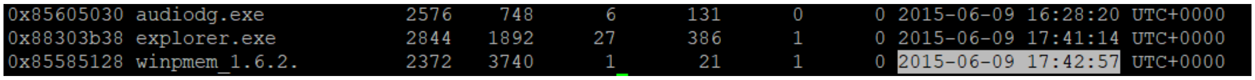


Honey Badger 1-7

30

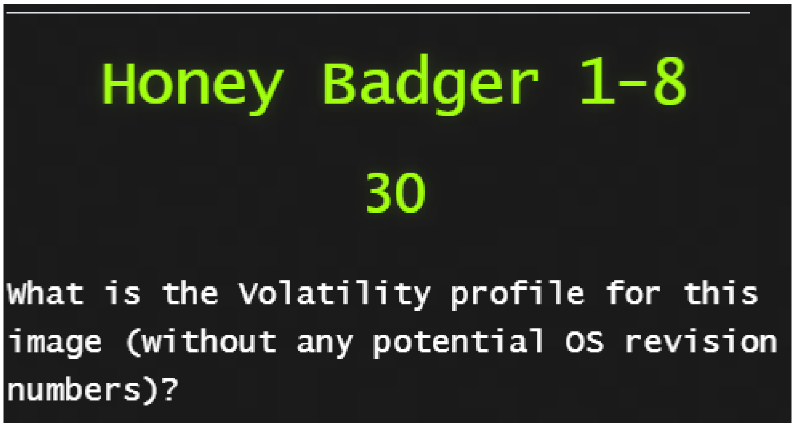
When was this memory image captured in Zulu time? FORMAT YYYY-MM-DD HH:MM:SS

For this we will look at the pslist plugin and find the pmem creation time



```
0x85605030 audiodg.exe            2576    748      6      131      0       0 2015-06-09 16:28:20 UTC+0000
0x88303b38 explorer.exe           2844   1892     27      386      1       0 2015-06-09 17:41:14 UTC+0000
0x85585128 winpmem_1.6.2.         2372   3740      1       21      1       0 2015-06-09 17:42:57 UTC+0000
```

Flag: 2015-06-09 17:42:57

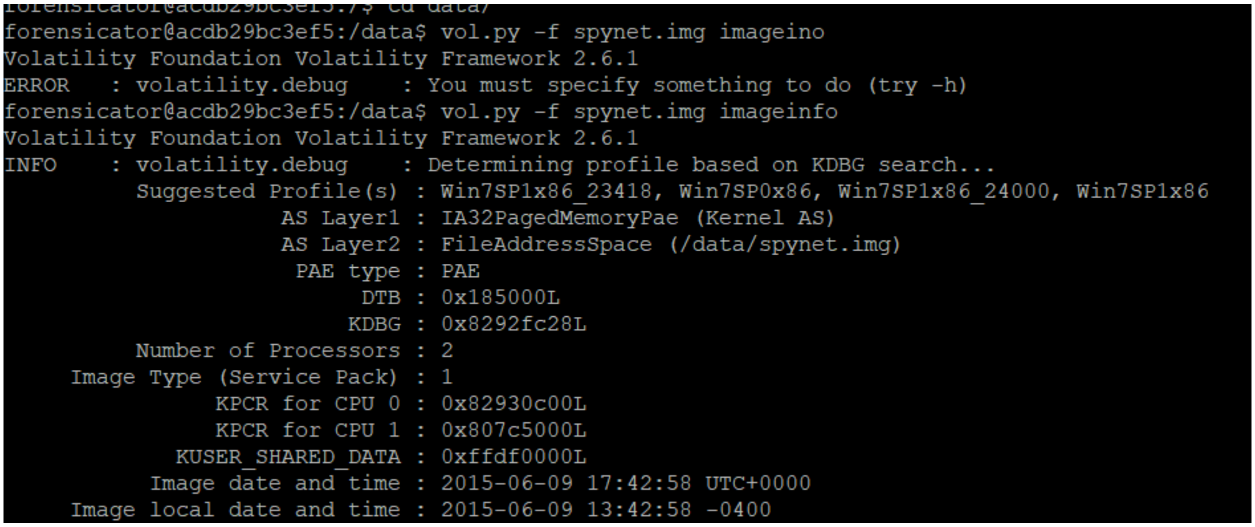


Honey Badger 1-8

30

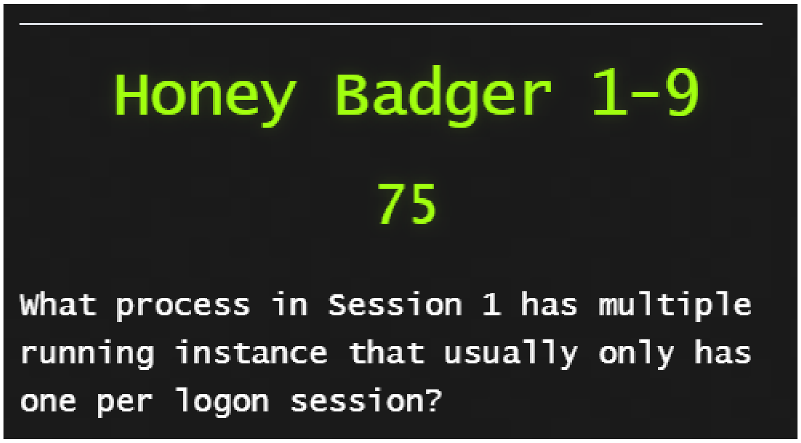
What is the Volatility profile for this image (without any potential OS revision numbers)?

Run the imageinfo to get the profile needed



```
forensicator@acdb29bc3ef5:/$ cd data/
forensicator@acdb29bc3ef5:/data$ vol.py -f spynet.img imageino
Volatility Foundation Volatility Framework 2.6.1
ERROR   : volatility.debug    : You must specify something to do (try -h)
forensicator@acdb29bc3ef5:/data$ vol.py -f spynet.img imageinfo
Volatility Foundation Volatility Framework 2.6.1
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : Win7SP1x86_23418, Win7SP0x86, Win7SP1x86_24000, Win7SP1x86
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace (/data/spynet.img)
                      PAE type : PAE
                           DTB : 0x185000L
                          KDBG : 0x8292fc28L
          Number of Processors : 2
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0x82930c00L
                KPCR for CPU 1 : 0x807c5000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2015-06-09 17:42:58 UTC+0000
     Image local date and time : 2015-06-09 13:42:58 -0400
```

Flag: Win7SP1x86

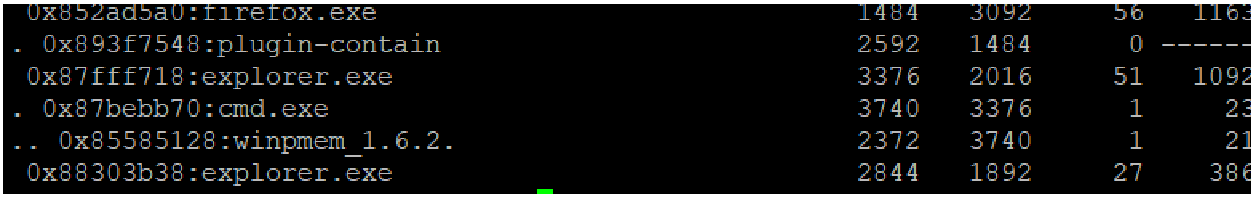


Honey Badger 1-9

75

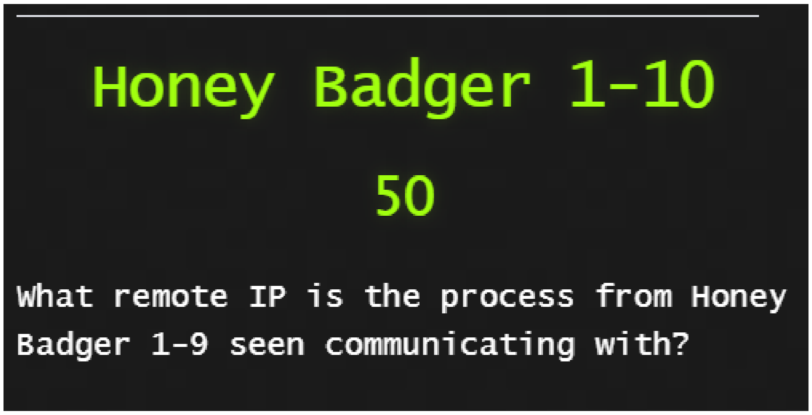
What process in Session 1 has multiple running instance that usually only has one per logon session?

Running the pstree and psxview plugins to find this answer



```
 0x852ad5a0:firefox.exe                                1484   3092      56   1163
. 0x893f7548:plugin-contain                            2592   1484       0 ------
 0x87fff718:explorer.exe                               3376   2016      51   1092
. 0x87bebb70:cmd.exe                                   3740   3376       1     23
.. 0x85585128:winpmem_1.6.2.                           2372   3740       1     21
 0x88303b38:explorer.exe                               2844   1892      27    386
```

Flag: explorer.exe



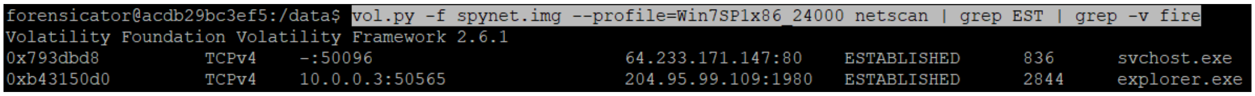
## Honey Badger 1-10

50

What remote IP is the process from Honey Badger 1-9 seen communicating with?

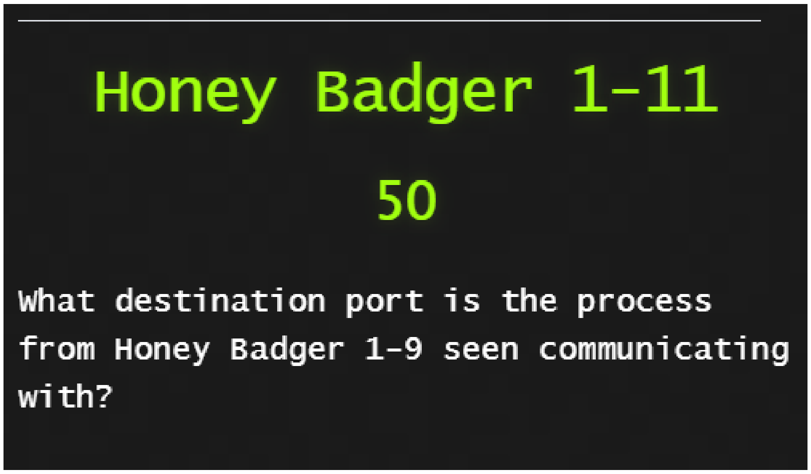
Run the following command to find the established connection from the explorer process

vol.py -f spynet.img --profile=Win7SP1x86_24000 netscan | grep EST | grep -v fire



```
forensicator@acdb29bc3ef5:/data$ vol.py -f spynet.img --profile=Win7SP1x86_24000 netscan | grep EST | grep -v fire
Volatility Foundation Volatility Framework 2.6.1
0x793dbd8          TCPv4    -:50096                          64.233.171.147:80    ESTABLISHED      836      svchost.exe
0xb43150d0         TCPv4    10.0.0.3:50565                   204.95.99.109:1980   ESTABLISHED      2844     explorer.exe
```

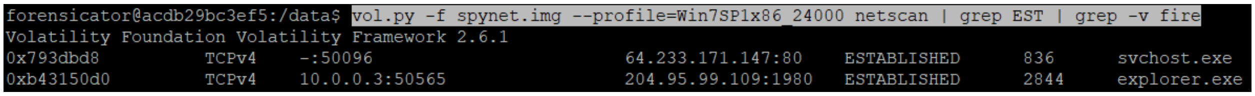
Flag: 204.95.99.109



## Honey Badger 1-11

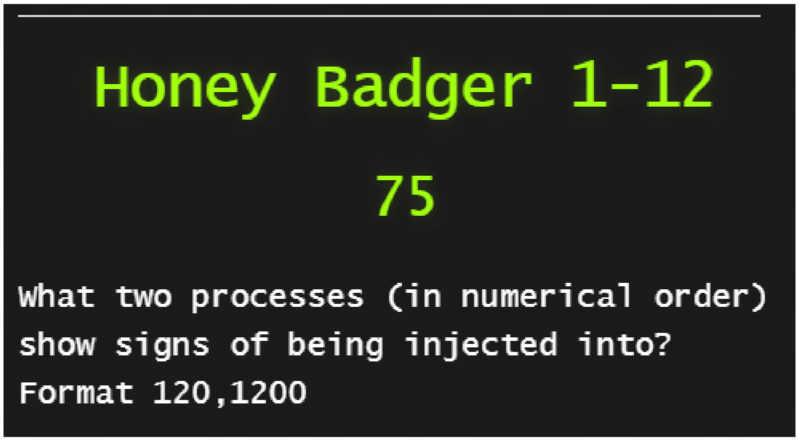
50

What destination port is the process from Honey Badger 1-9 seen communicating with?



```
forensicator@acdb29bc3ef5:/data$ vol.py -f spynet.img --profile=Win7SP1x86_24000 netscan | grep EST | grep -v fire
Volatility Foundation Volatility Framework 2.6.1
0x793dbd8          TCPv4    -:50096                          64.233.171.147:80    ESTABLISHED      836      svchost.exe
0xb43150d0         TCPv4    10.0.0.3:50565                   204.95.99.109:1980   ESTABLISHED      2844     explorer.exe
```

Flag 1980



Honey Badger 1-12

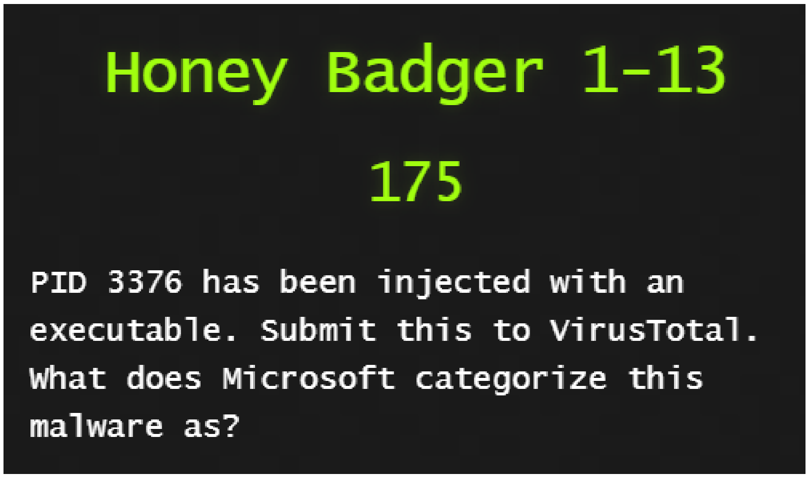
75

What two processes (in numerical order) show signs of being injected into? Format 120,1200

Run the following command to get a list of pid with malfind:

```
vol.py -f spynet.img --profile=Win7SP1x86_24000 malfind | grep -i pid | grep -v fire | sort -u | more
```

Flag: 2844,3376



Honey Badger 1-13

175

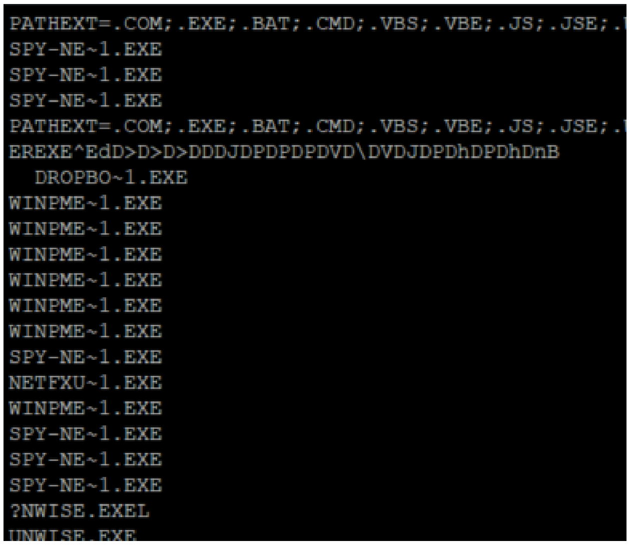
PID 3376 has been injected with an executable. Submit this to VirusTotal. What does Microsoft categorize this malware as?

For this we are going to dump the memory of processed 3376, vol.py -f spynet.img --profile=Win7SP1x86_24000 memdump -p 3376 -D /tmp/3376mem/

Then we will create a strings file: strings /tmp/3376mem/3376.dmp > /tmp/3376mem/strings

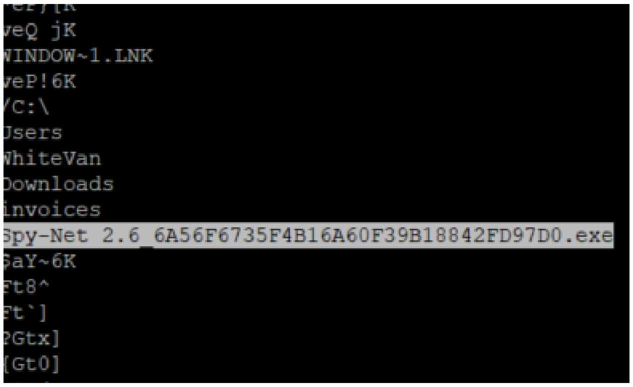
From here we will cat the file and look through it or grep for EXE

Right after the path when grepping EXE we see the short notation of the possible program



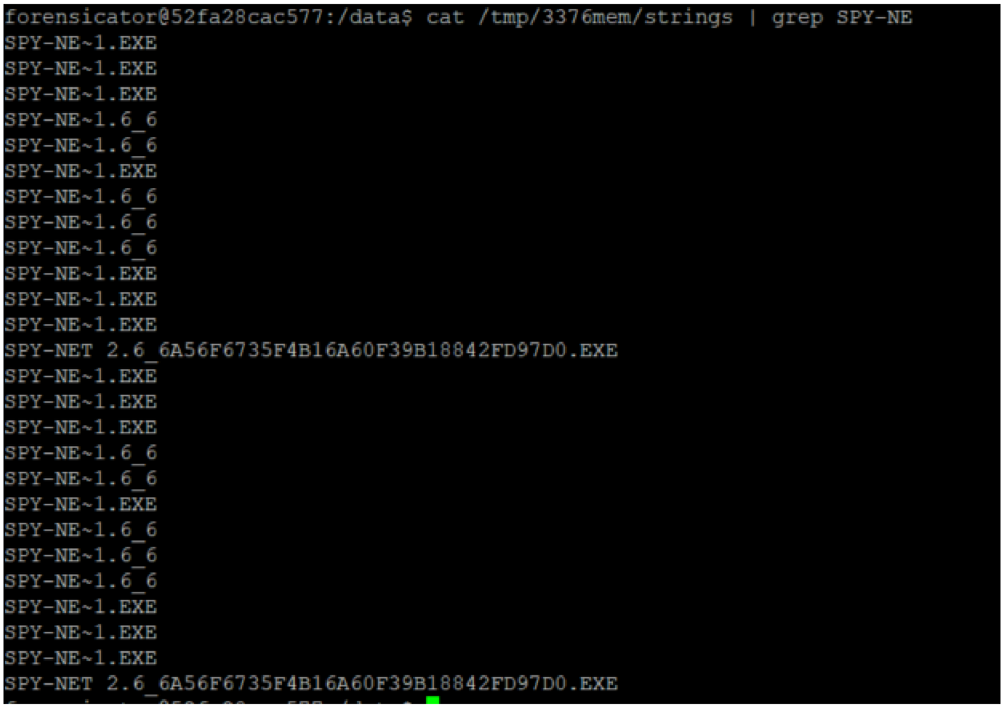
```
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.
SPY-NE~1.EXE
SPY-NE~1.EXE
SPY-NE~1.EXE
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.
EREXE^EdD>D>D>DDDJDPDPDPDVD\DVDJDPDhDPDhDnB
  DROPBO~1.EXE
WINPME~1.EXE
WINPME~1.EXE
WINPME~1.EXE
WINPME~1.EXE
WINPME~1.EXE
WINPME~1.EXE
SPY-NE~1.EXE
NETFXU~1.EXE
WINPME~1.EXE
SPY-NE~1.EXE
SPY-NE~1.EXE
SPY-NE~1.EXE
?NWISE.EXEL
UNWISE.EXE
```

You could pivot on this if you recognized it, otherwise you can keep scrolling though until you come across Spy-Net



```
veQ jK
WINDOW~1.LNK
veP!6K
/C:\
Users
WhiteVan
Downloads
invoices
Spy-Net 2.6_6A56F6735F4B16A60F39B18842FD97D0.exe
$aY~6K
Ft8^
Ft`]
PGtx]
[Gt0]
```

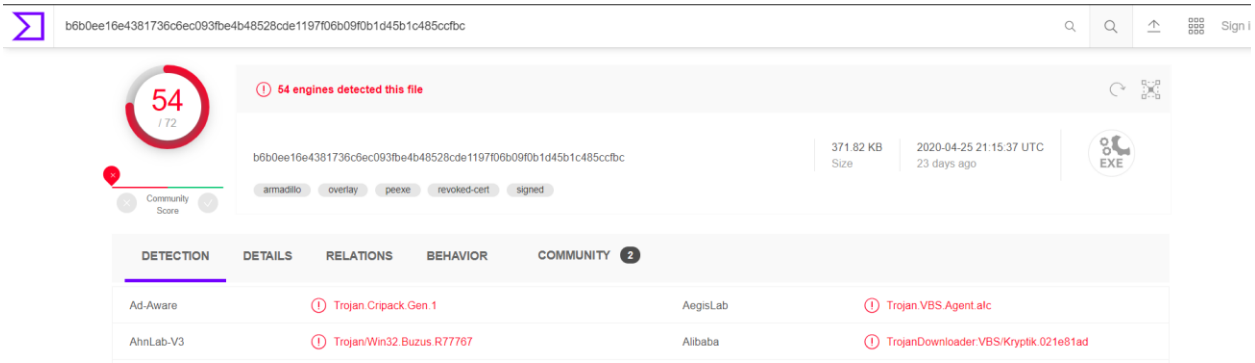
If you were to move from the SPY-NE you would get this:



```
forensicator@52fa28cac577:/data$ cat /tmp/3376mem/strings | grep SPY-NE
SPY-NE~1.EXE
SPY-NE~1.EXE
SPY-NE~1.EXE
SPY-NE~1.6_6
SPY-NE~1.6_6
SPY-NE~1.EXE
SPY-NE~1.6_6
SPY-NE~1.6_6
SPY-NE~1.6_6
SPY-NE~1.EXE
SPY-NE~1.EXE
SPY-NE~1.EXE
SPY-NET 2.6_6A56F6735F4B16A60F39B18842FD97D0.EXE
SPY-NE~1.EXE
SPY-NE~1.EXE
SPY-NE~1.EXE
SPY-NE~1.6_6
SPY-NE~1.6_6
SPY-NE~1.EXE
SPY-NE~1.6_6
SPY-NE~1.6_6
SPY-NE~1.6_6
SPY-NE~1.EXE
SPY-NE~1.EXE
SPY-NE~1.EXE
SPY-NET 2.6_6A56F6735F4B16A60F39B18842FD97D0.EXE
```

Our md5 is 6A56F6735F4B16A60F39B18842FD97D0

Post that into virus total to get the following info:



b6b0ee16e4381736c6ec093fbe4b48528cde1197f06b09f0b1d45b1c485ccfbc

54 / 72

54 engines detected this file

b6b0ee16e4381736c6ec093fbe4b48528cde1197f06b09f0b1d45b1c485ccfbc

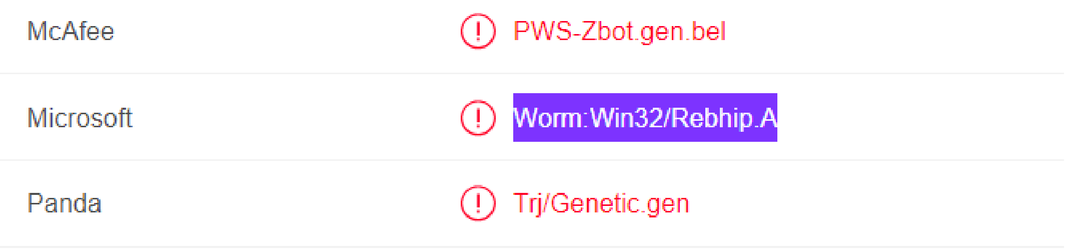
armadillo overlay peexe revoked-cert signed

371.82 KB Size

2020-04-25 21:15:37 UTC 23 days ago

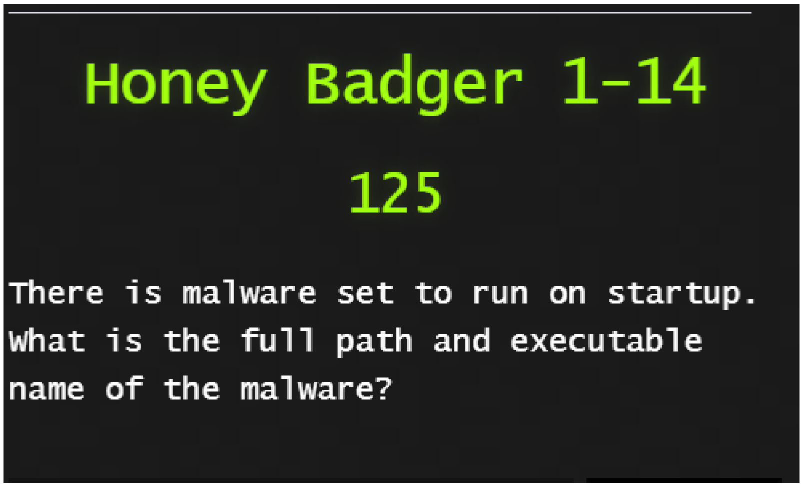
| | | | |
|---|---|---|---|
| Ad-Aware | Trojan.Cripack.Gen.1 | AegisLab | Trojan.VBS.Agent.alc |
| AhnLab-V3 | Trojan/Win32.Buzus.R77767 | Alibaba | TrojanDownloader:VBS/Kryptik.021e81ad |

Microsoft marks this as the following:



| | |
|---|---|
| McAfee | ❗ PWS-Zbot.gen.bel |
| Microsoft | ❗ Worm:Win32/Rebhip.A |
| Panda | ❗ Trj/Genetic.gen |

Flag: Worm:Win32/Rebhip.A

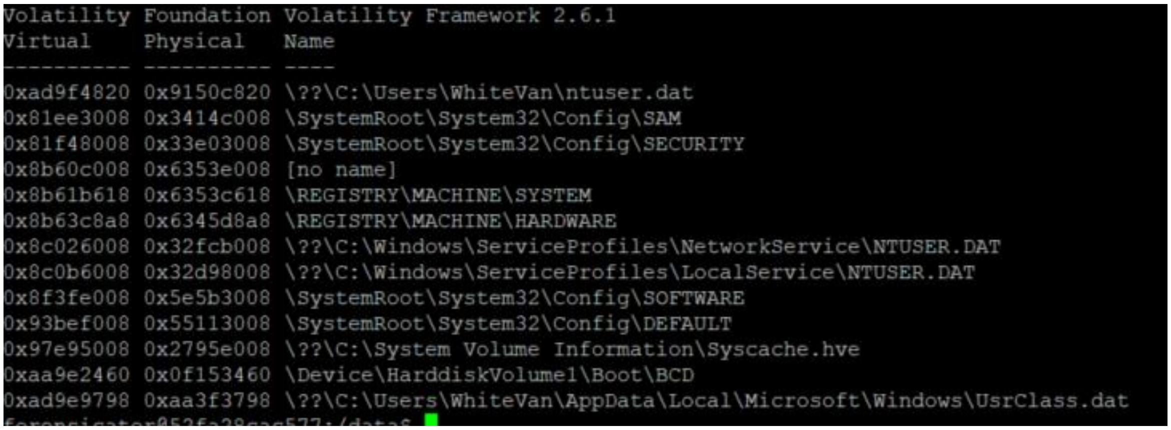


Honey Badger 1-14

125

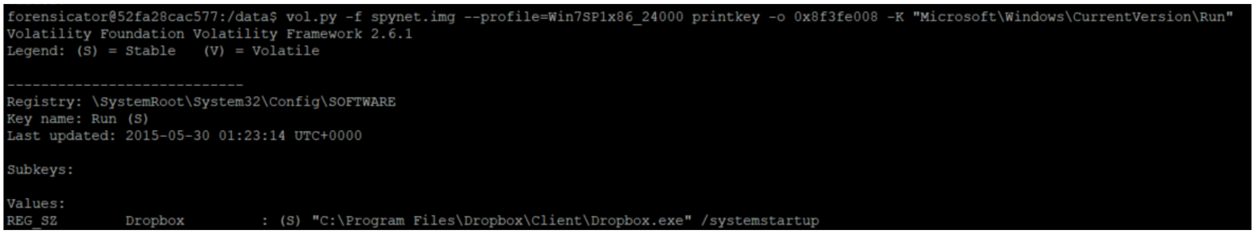
There is malware set to run on startup. What is the full path and executable name of the malware?

For this one we will dig into the registry of the image: vol.py -f spynet.img --profile=Win7SP1x86_24000 hivelist



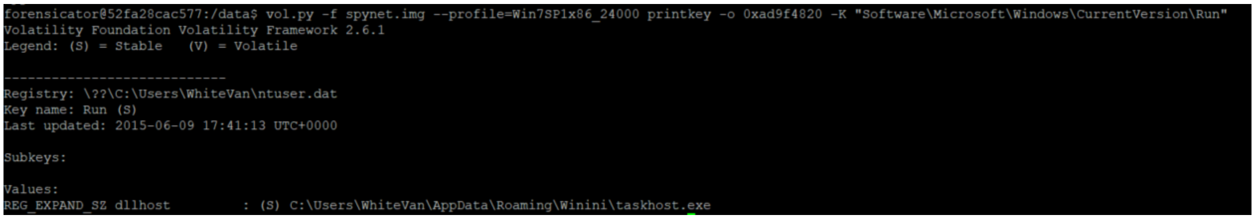
```
Volatility Foundation Volatility Framework 2.6.1
Virtual    Physical   Name
---------- ---------- ----
0xad9f4820 0x9150c820 \??\C:\Users\WhiteVan\ntuser.dat
0x81ee3008 0x3414c008 \SystemRoot\System32\Config\SAM
0x81f48008 0x33e03008 \SystemRoot\System32\Config\SECURITY
0x8b60c008 0x6353e008 [no name]
0x8b61b618 0x6353c618 \REGISTRY\MACHINE\SYSTEM
0x8b63c8a8 0x6345d8a8 \REGISTRY\MACHINE\HARDWARE
0x8c026008 0x32fcb008 \??\C:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
0x8c0b6008 0x32d98008 \??\C:\Windows\ServiceProfiles\LocalService\NTUSER.DAT
0x8f3fe008 0x5e5b3008 \SystemRoot\System32\Config\SOFTWARE
0x93bef008 0x55113008 \SystemRoot\System32\Config\DEFAULT
0x97e95008 0x2795e008 \??\C:\System Volume Information\Syscache.hve
0xaa9e2460 0x0f153460 \Device\HarddiskVolume1\Boot\BCD
0xad9e9798 0xaa3f3798 \??\C:\Users\WhiteVan\AppData\Local\Microsoft\Windows\UsrClass.dat
forensicator@52fa28cac577:/data$
```

We get the system hive offset as 0x8f3fe008, after looking there we only get a dropbox autorun, this is not the answer we are looking for.



```
forensicator@52fa28cac577:/data$ vol.py -f spynet.img --profile=Win7SP1x86_24000 printkey -o 0x8f3fe008 -K "Microsoft\Windows\CurrentVersion\Run"
Volatility Foundation Volatility Framework 2.6.1
Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: \SystemRoot\System32\Config\SOFTWARE
Key name: Run (S)
Last updated: 2015-05-30 01:23:14 UTC+0000

Subkeys:

Values:
REG_SZ        Dropbox         : (S) "C:\Program Files\Dropbox\Client\Dropbox.exe" /systemstartup
```

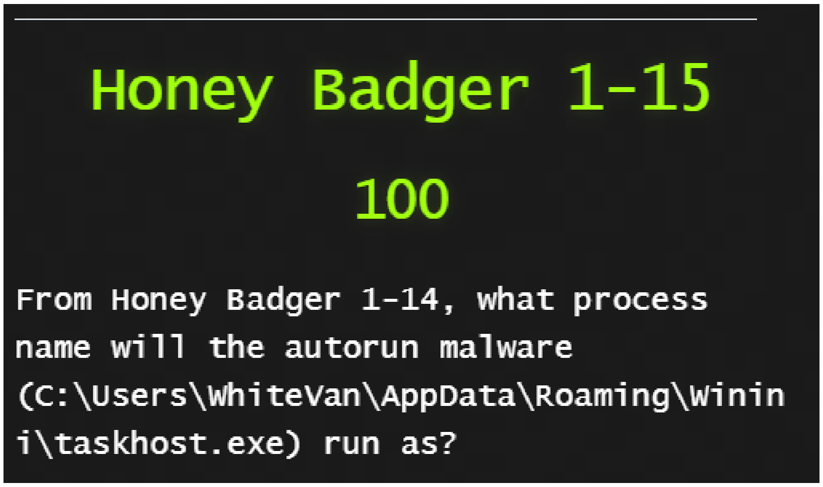
Next we will look at the whitevan user hive to see if there are any autoruns there.



```
forensicator@52fa28cac577:/data$ vol.py -f spynet.img --profile=Win7SP1x86_24000 printkey -o 0xad9f4820 -K "Software\Microsoft\Windows\CurrentVersion\Run"
Volatility Foundation Volatility Framework 2.6.1
Legend: (S) = Stable   (V) = Volatile

----------------------------
Registry: \??\C:\Users\WhiteVan\ntuser.dat
Key name: Run (S)
Last updated: 2015-06-09 17:41:13 UTC+0000

Subkeys:

Values:
REG_EXPAND_SZ dllhost         : (S) C:\Users\WhiteVan\AppData\Roaming\Winini\taskhost.exe
```

That is an odd location for taskhost.exe to be running from as it is not in system32 folder.

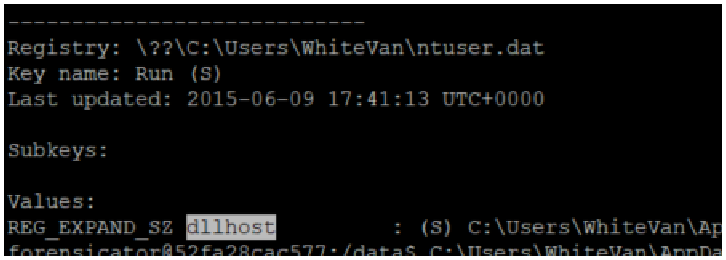
Flag: C:\Users\WhiteVan\AppData\Roaming\Winini\taskhost.exe



## Honey Badger 1-15

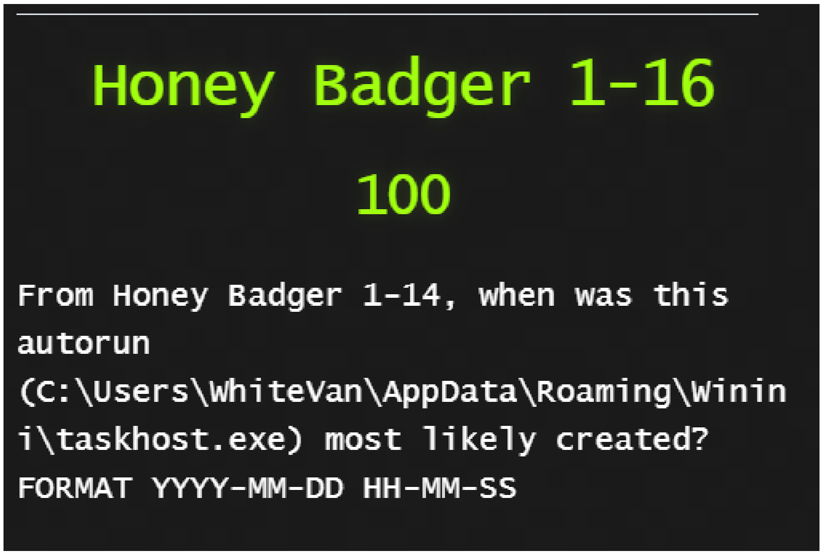
100

From Honey Badger 1-14, what process name will the autorun malware (C:\Users\WhiteVan\AppData\Roaming\Winini\taskhost.exe) run as?



```
----------------------------
Registry: \??\C:\Users\WhiteVan\ntuser.dat
Key name: Run (S)
Last updated: 2015-06-09 17:41:13 UTC+0000

Subkeys:

Values:
REG_EXPAND_SZ dllhost          : (S) C:\Users\WhiteVan\Ap
forensicator@52fa28cac577:/data$ C:\Users\WhiteVan\AppDa
```

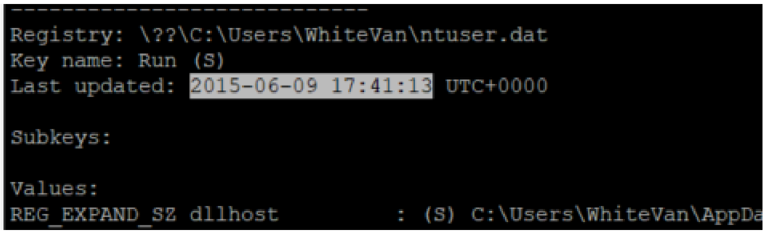
Flag: dllhost



## Honey Badger 1-16

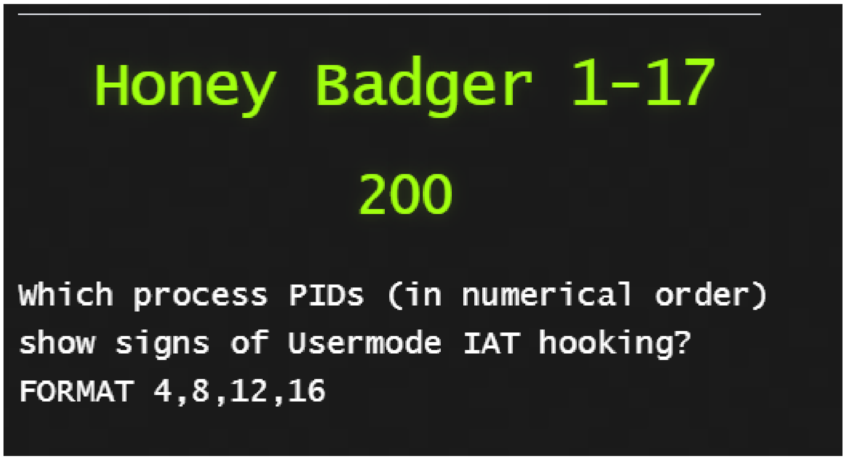
100

From Honey Badger 1-14, when was this autorun (C:\Users\WhiteVan\AppData\Roaming\Winini\taskhost.exe) most likely created? FORMAT YYYY-MM-DD HH-MM-SS



```
----------------------------
Registry: \??\C:\Users\WhiteVan\ntuser.dat
Key name: Run (S)
Last updated: 2015-06-09 17:41:13 UTC+0000

Subkeys:

Values:
REG_EXPAND_SZ dllhost         : (S) C:\Users\WhiteVan\AppDa
```

Flag: 2015-06-09 17:41:13



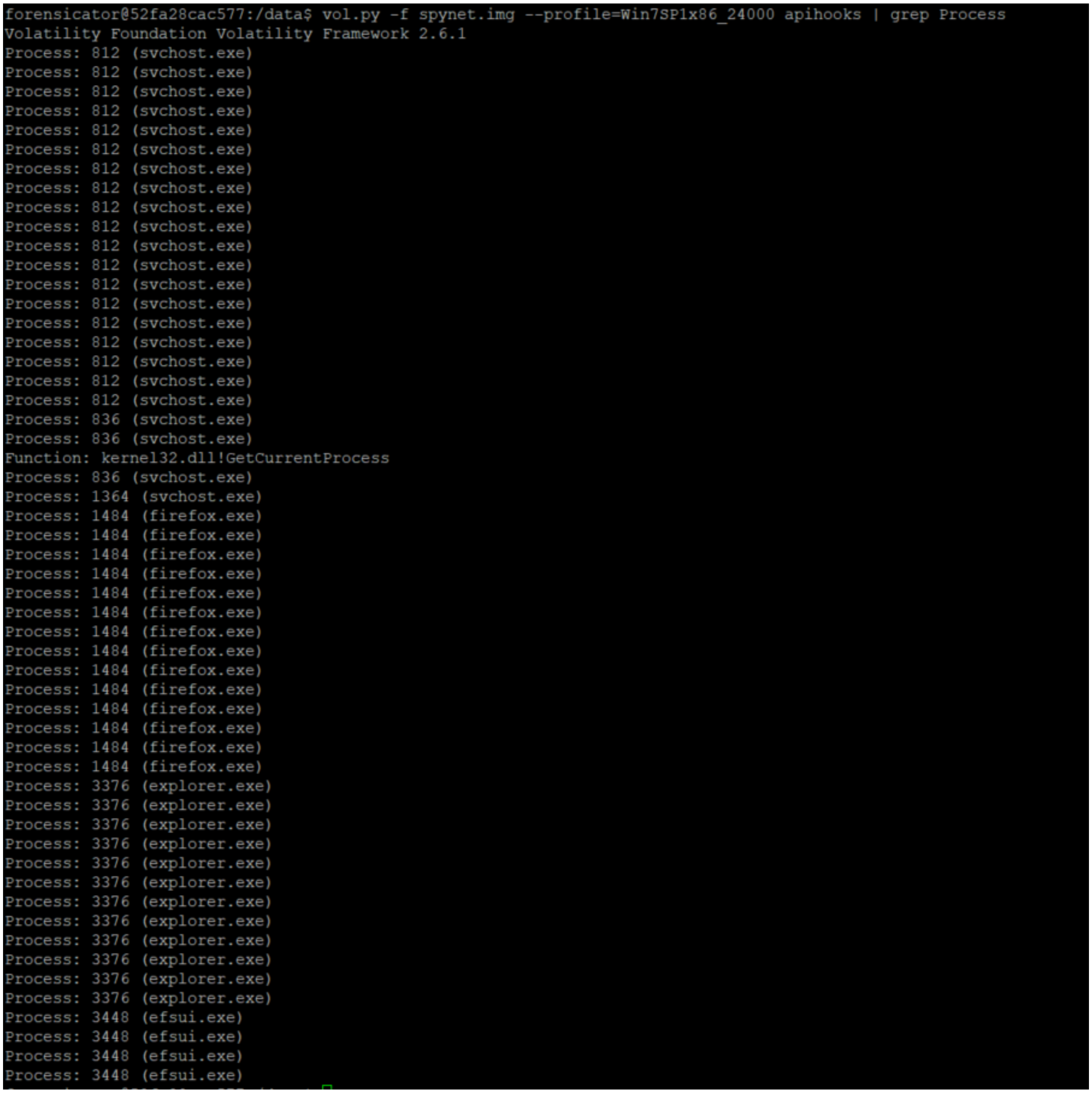
Honey Badger 1-17

200

Which process PIDs (in numerical order) show signs of Usermode IAT hooking? FORMAT 4,8,12,16

For this one we will run the apihooks plugin, but this will take some time to run. Grepping for Process will give us only the process information that we need right now.

We see



```
forensicator@52fa28cac577:/data$ vol.py -f spynet.img --profile=Win7SP1x86_24000 apihooks | grep Process
Volatility Foundation Volatility Framework 2.6.1
Process: 812 (svchost.exe)
Process: 812 (svchost.exe)
Process: 812 (svchost.exe)
Process: 812 (svchost.exe)
Process: 812 (svchost.exe)
Process: 812 (svchost.exe)
Process: 812 (svchost.exe)
Process: 812 (svchost.exe)
Process: 812 (svchost.exe)
Process: 812 (svchost.exe)
Process: 812 (svchost.exe)
Process: 812 (svchost.exe)
Process: 812 (svchost.exe)
Process: 812 (svchost.exe)
Process: 812 (svchost.exe)
Process: 812 (svchost.exe)
Process: 812 (svchost.exe)
Process: 812 (svchost.exe)
Process: 812 (svchost.exe)
Process: 836 (svchost.exe)
Process: 836 (svchost.exe)
Function: kernel32.dll!GetCurrentProcess
Process: 836 (svchost.exe)
Process: 1364 (svchost.exe)
Process: 1484 (firefox.exe)
Process: 1484 (firefox.exe)
Process: 1484 (firefox.exe)
Process: 1484 (firefox.exe)
Process: 1484 (firefox.exe)
Process: 1484 (firefox.exe)
Process: 1484 (firefox.exe)
Process: 1484 (firefox.exe)
Process: 1484 (firefox.exe)
Process: 1484 (firefox.exe)
Process: 1484 (firefox.exe)
Process: 1484 (firefox.exe)
Process: 1484 (firefox.exe)
Process: 1484 (firefox.exe)
Process: 3376 (explorer.exe)
Process: 3376 (explorer.exe)
Process: 3376 (explorer.exe)
Process: 3376 (explorer.exe)
Process: 3376 (explorer.exe)
Process: 3376 (explorer.exe)
Process: 3376 (explorer.exe)
Process: 3376 (explorer.exe)
Process: 3376 (explorer.exe)
Process: 3376 (explorer.exe)
Process: 3376 (explorer.exe)
Process: 3376 (explorer.exe)
Process: 3448 (efsui.exe)
Process: 3448 (efsui.exe)
Process: 3448 (efsui.exe)
Process: 3448 (efsui.exe)
```

We want all of the processes besides Firefox to get our answer

Flag: 812,836,1364,3376,3448