



Investigating Network Activities

Networking

- Most malware has some sort of network activity
- Malware connects to Command and Control server (C2) to
 - *download additional components*
 - *report the infection to the attacker.*
 - *receive commands*
 - *exfiltrate data*
- Malwares spread to other machines on the network

Volatility Network Plugins

Below plugins work on Pre Vista Systems:

- **connections** - print list of open connections [Windows XP & 2003 Only]
- **connscan** - uses pool scanning to display network connections
- **sockets** - print list of open sockets
- **sockscan** - uses pools scanning to display open sockets

Below plugin can be used on Vista or later systems:

- **netscan** - displays both connections and sockets

Displaying Network Connections using *connections* and *connscan* plugin

The screenshot shows the network connections associated with pid **1560** which is associated with process **IEXPLORE.exe**

```
root@kratos:~/Volatility# python vol.py -f xrat.vmem connections
Volatility Foundation Volatility Framework 2.5
Offset(V)  Local Address          Remote Address          Pid
-----
0x816fe240 192.168.1.100:1033     192.168.1.3:81         1560 ←
root@kratos:~/Volatility#
root@kratos:~/Volatility# python vol.py -f xrat.vmem connscan
Volatility Foundation Volatility Framework 2.5
Offset(P)  Local Address          Remote Address          Pid
-----
0x018fe240 192.168.1.100:1033     192.168.1.3:81         1560 ←
root@kratos:~/Volatility#
root@kratos:~/Volatility# python vol.py -f xrat.vmem pslist -p 1560
Volatility Foundation Volatility Framework 2.5
Offset(V)  Name                   PID  PPID  Thds  Hnds  Sess  Wow64  Start
-----
0x81414438 IEXPLORE.EXE          1560 1924  7     182   0     0     2016-04-30 17:41:34
UTC+0000
root@kratos:~/Volatility#
```

Displaying Listening Sockets (sockets and sockscan)

```
root@kratos:~/Volatility# python vol.py -f xrat.vmem sockets
```

```
Volatility Foundation Volatility Framework 2.5
```

| Offset(V) | PID | Port | Proto | Protocol | Address | Create Time |
|------------|------|------|-------|----------|---------------|------------------------------|
| 0x815db0e0 | 4 | 0 | 47 | GRE | 0.0.0.0 | 2016-04-30 17:41:36 UTC+0000 |
| 0x81877670 | 892 | 123 | 17 | UDP | 192.168.1.100 | 2016-04-03 18:44:55 UTC+0000 |
| 0x81624d28 | 4 | 1034 | 6 | TCP | 0.0.0.0 | 2016-04-30 17:41:36 UTC+0000 |
| 0x8171c868 | 584 | 500 | 17 | UDP | 0.0.0.0 | 2016-04-03 18:44:55 UTC+0000 |
| 0x818fd770 | 4 | 445 | 6 | TCP | 0.0.0.0 | 2016-04-03 18:44:52 UTC+0000 |
| 0x8151c118 | 832 | 135 | 6 | TCP | 0.0.0.0 | 2016-04-03 18:44:53 UTC+0000 |
| 0x81700b38 | 1684 | 1025 | 6 | TCP | 127.0.0.1 | 2016-04-03 18:44:56 UTC+0000 |
| 0x81509ce8 | 1064 | 1900 | 17 | UDP | 192.168.1.100 | 2016-04-03 18:44:56 UTC+0000 |
| 0x814bb7e0 | 584 | 0 | 255 | Reserved | 0.0.0.0 | 2016-04-03 18:44:55 UTC+0000 |
| 0x81704070 | 892 | 123 | 17 | UDP | 127.0.0.1 | 2016-04-03 18:44:55 UTC+0000 |

```
root@kratos:~/Volatility# python vol.py -f xrat.vmem sockscan
```

```
Volatility Foundation Volatility Framework 2.5
```

| Offset(P) | PID | Port | Proto | Protocol | Address | Create Time |
|------------|------|------|-------|----------|---------------|------------------------------|
| 0x01552d00 | 584 | 4500 | 17 | UDP | 0.0.0.0 | 2016-04-03 18:44:55 UTC+0000 |
| 0x016bb7e0 | 584 | 0 | 255 | Reserved | 0.0.0.0 | 2016-04-03 18:44:55 UTC+0000 |
| 0x01709ce8 | 1064 | 1900 | 17 | UDP | 192.168.1.100 | 2016-04-03 18:44:56 UTC+0000 |
| 0x0171c118 | 832 | 135 | 6 | TCP | 0.0.0.0 | 2016-04-03 18:44:53 UTC+0000 |
| 0x017db0e0 | 4 | 0 | 47 | GRE | 0.0.0.0 | 2016-04-30 17:41:36 UTC+0000 |
| 0x01822990 | 4 | 139 | 6 | TCP | 192.168.1.100 | 2016-04-03 18:44:52 UTC+0000 |
| 0x01822b98 | 4 | 137 | 17 | UDP | 192.168.1.100 | 2016-04-03 18:44:52 UTC+0000 |
| 0x01824d28 | 4 | 1034 | 6 | TCP | 0.0.0.0 | 2016-04-30 17:41:36 UTC+0000 |

Displaying connections and sockets using *netscan* plugin

```
root@kratos:~/Volatility# python vol.py -f kuluoz.vmem --profile=Win7SP0x86 netscan
```

```
Volatility Foundation Volatility Framework 2.5
```

| Offset(P) | Proto | Local Address | Foreign Address | State |
|------------|-------------|-------------------------------------|-------------------|-------------|
| Pid | Owner | Created | | |
| 0x171b3480 | System | TCPv4 192.168.1.60:139 | 0.0.0.0:0 | LISTENING |
| 4 | | | | |
| 0x17e5d920 | System | TCPv4 0.0.0.0:49152 | 0.0.0.0:0 | LISTENING |
| 396 | wininit.exe | | | |
| 0x1dc052d0 | svchost.exe | UDPv4 0.0.0.0:5355 | *:* | |
| 1152 | svchost.exe | 2016-05-11 06:35:33 UTC+0000 | | |
| 0x1dc05388 | svchost.exe | UDPv4 0.0.0.0:0 | *:* | |
| 1152 | svchost.exe | 2016-05-11 06:35:33 UTC+0000 | | |
| 0x1dc05388 | svchost.exe | UDPv6 :::0 | *:* | |
| 916 | svchost.exe | TCPv6 :::49154 | :::0 | LISTENING |
| 0x1e1caa80 | svchost.exe | TCPv4 0.0.0.0:49156 | 0.0.0.0:0 | LISTENING |
| 916 | svchost.exe | TCPv4 0.0.0.0:49156 | 0.0.0.0:0 | LISTENING |
| 0x1e1e5430 | lsass.exe | TCPv6 :::49156 | :::0 | LISTENING |
| 504 | lsass.exe | TCPv4 192.168.1.60:49159 | 1.234.20.244:8080 | ESTABLISHED |
| 0x1f1fe440 | svchost.exe | UDPv6 fe80::7ce6:2db4:2925:6273:546 | *:* | |
| 3056 | svchost.exe | 2016-05-11 06:35:41 UTC+0000 | | |
| 0x1f57d008 | svchost.exe | | | |
| 764 | svchost.exe | | | |

Lab 13.1: The Case of Spybot (contd.)

Use the memory image (***spybot.vmem***)

- Can you identify the C2 ip address?
- What port/protocol is the malware using for communication?