Lokibot Malware Analysis Report

Summary of Analysis

Analysed malware sample is a Lokibot variant using which as information stealer. It uses some anti-analysis techniques like garbage-code(E4 opcode) for anti-disassembly, some API functions for anti-debugging. Also it has process injection, network communication capabilities. It tries to steal FTP, email and browser credentials.

Identification

MD5	037b874a119a7cd0e00a3c971dd3298a
SHA1	dd5237c02f7f5711e67149c47762b41df081cc0c
SSDEEP	1536:SzvQSZpGS4/31A6mQgL2eYCGDwRcMkVQd8YhY0/ EqoIzmd:pSHIG6mQwGmfOQd8YhY0/EpUG
File Type	PE32
File Size	648.00 KB

Characteristics

Malware can uses CreateToolhelp32Snapshot, IsDebuggerPresent, ChechRemoteDebugger and NtQueryInformationProcess to prevent debugging. Therefore we can trace these API functions in debugging. Malware was resolving API functions as dynamically.

```
kernel32_CreateProcessW
kernel32_GetThreadContext
ntdll_NtClose
ntdll_NtUnmapViewOfSection
kernel32_SizeofResource
kernel32_GetFileAttributesW
kernel32_SetFileAttributesW
kernel32_SetFilePointer
kernel32_WideCharToMultiByte
kernel32_VirtualFree
kernel32_GetModuleHandleA
kernel32_GetProcAddress
kernel32_VirtualAlloc
kernel32_SetThreadContext
kernel32_LoadResource
ntdll_NtQueryInformationProcess
kernel32_ResumeThread
```

```
offset kernel32_Sleep
offset kernel32_CreateToolhelp32Snapshot
offset kernel32_Process32FirstW
offset kernel32_Process32NextW
offset kernel32_ReadProcessMemory
offset kernel32_WriteProcessMemory
offset kernel32_VirtualAllocEx
offset kernel32_GetSystemDirectoryW
offset kernel32_CopyFileW
offset kernel32_DeleteFileW
offset kernel32_CreateDirectoryW
offset kernel32_WriteFile
offset kernel32_CreateFileW
offset kernel32_CloseHandle
offset kernel32_GetFileSize
offset kernel32_ReadFile
offset kernel32_LoadLibraryA
```

Malware was checking firstly malware names. Checked malware sample names are *sandbox, sample, self, virus*.

```
0018FA10  00 00 00 00 00 00 00 00  00 00 00 00 73 00 61 00  ............s.a.
0018FA20  6E 00 64 00 62 00 6F 00  78 00 00 00 6D 00 61 00  n.d.b.o.x...m.a.
0018FA30  6C 00 77 00 61 00 72 00  65 00 00 00 73 00 61 00  l.w.a.r.e...s.a.
0018FA40  6D 00 70 00 6C 00 65 00  00 00 00 00 76 00 69 00  m.p.l.e.....v.i.
0018FA50  72 00 75 00 73 00 00 00  73 00 65 00 6C 00 66 00  r.u.s...s.e.l.f.
0018FA60  2E 00 00 00 10 FF 18 00  35 68 D9 01 00 00 2E 00  .....ÿ..5hÙ.....
```

Another anti-debugging check perform for avast antivirus. Checked loaded process names are *avastsvc.exe, avastui.exe, avgsvc.exe, iavgui.exe*.

```
0018FA00  00 00 00 00 61 00 76 00  61 00 73 00 74 00 73 00  ....a.v.a.s.t.s.
0018FA10  76 00 63 00 2E 00 65 00  78 00 65 00 00 00 61 00  v.c...e.x.e...a.
0018FA20  61 00 76 00 61 00 73 00  74 00 75 00 69 00 2E 00  a.v.a.s.t.u.i...
0018FA30  65 00 78 00 65 00 00 00  61 00 76 00 67 00 73 00  e.x.e...a.v.g.s.
0018FA40  76 00 63 00 2E 00 65 00  78 00 65 00 00 00 69 00  v.c...e.x.e...i.
0018FA50  61 00 76 00 67 00 75 00  69 00 2E 00 65 00 78 00  a.v.g.u.i...e.x.
0018FA60  65 00 00 00 10 FF 18 00  2A 62 D9 01 00 00 2E 00  e....ÿ..*bÙ.....
```

Another check was performing to detect debugging tools. Checked debugging process names are *procexp64.exe*, *procmon64.exe*, *procmon.exe*, *ollydbg.exe*, *procexp.exe*, *windbg.exe*.

```
00 00 00 00 70 00 72 00  ............p.r.
70 00 36 00 34 00 2E 00  o.c.e.x.p.6.4...
70 00 72 00 6F 00 63 00  e.x.e...p.r.o.c.
34 00 2E 00 65 00 78 00  m.o.n.6.4...e.x.
6F 00 63 00 6D 00 6F 00  e...p.r.o.c.m.o.
65 00 00 00 6F 00 6C 00  n...e.x.e...o.l.
67 00 2E 00 65 00 78 00  l.y.d.b.g...e.x.
6F 00 63 00 65 00 78 00  e...p.r.o.c.e.x.
65 00 00 00 77 00 69 00  p...e.x.e...w.i.
2E 00 65 00 78 00 65 00  n.d.b.g...e.x.e.
```

Also malware checks both 0x1F and 0x1E QueryInformationClass.

```
.D96D23:                                ; CODE XREF: get_str_len+2109↑j
   call    dword ptr [esi+8Ch]          ; NtQueryInformationProcess
   cmp     [ebp-4], edi                 ; Compare Two Operands
   jmp     loc_1D980D9                  ; Jump
```

```
0018FA48  FFFFFFFF
0018FA4C  0000001F
```

(ProcessDebugObjectHandle)

```
loc_1D93A5D:                            ; CODE XREF: get_str_len:loc_
        call    dword ptr [esi+8Ch]     ; NtQueryInformationProcess
        test    eax, eax                ; Logical Compare
        jns     loc_1D96253             ; Jump if Not Sign (SF=0)
        jmp     loc_1D965C6             ; Jump
```

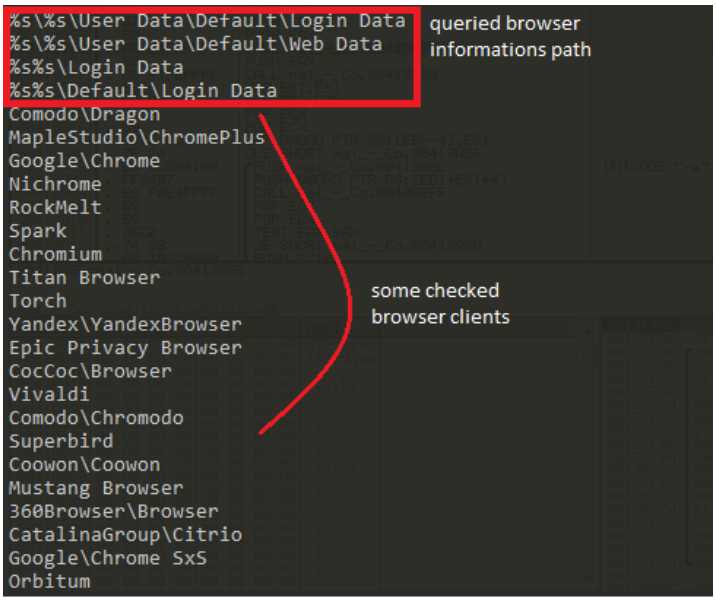
```
0018FA48  FFFFFFFF
0018FA4C  0000001E
```

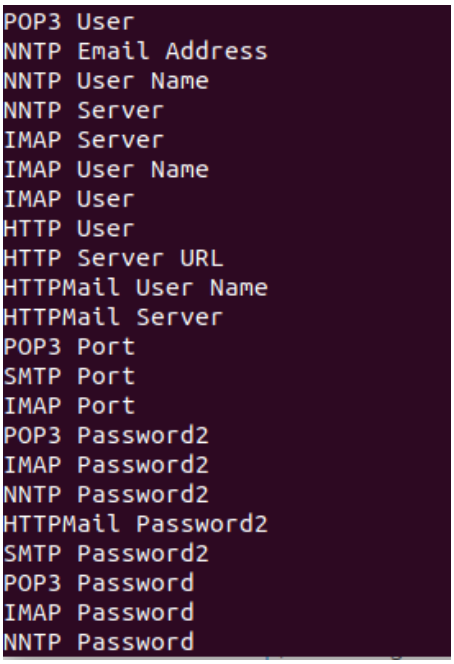
(ProcessDebugFlags)

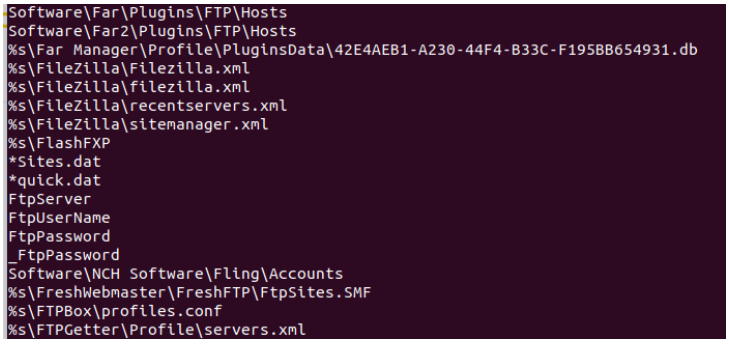
Malware was using process injection technique to install itself with same name. After injection it immediately terminated itself parent process and delete itself from harddisk.

malw.exe	2400	68,61
ConEmu.exe	2892	0,16
ConEmuC.exe	2524	0,07
cmd.exe	2616	
osk.exe	3548	
malw.exe	1548	

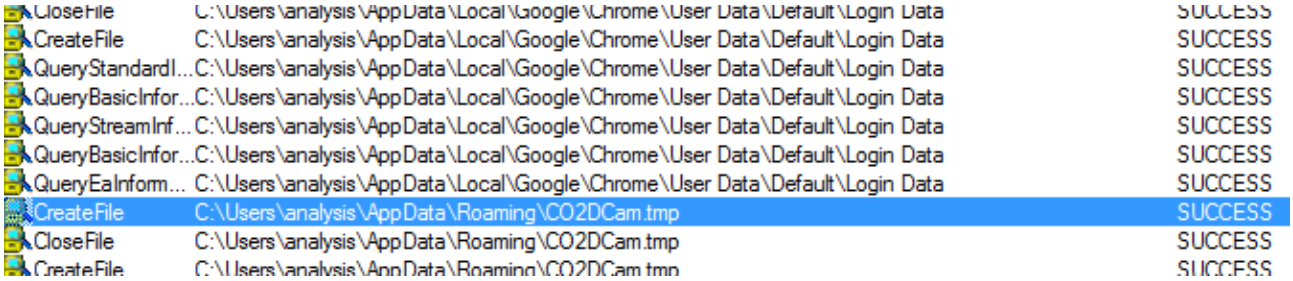
Unpacked malware has some suspicious strings indicates of author's purpose.



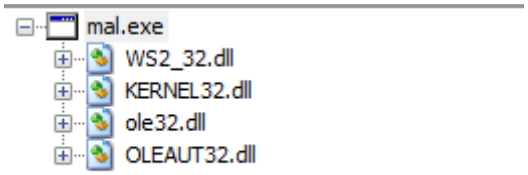


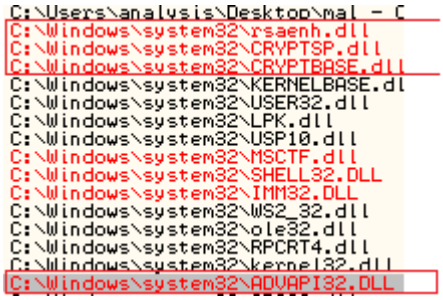


If you login any sites using identified browsers it looks like to steals your web credential and data.



Unpacked malware imports some libraries but it resolve dynamically which other important DLLs.

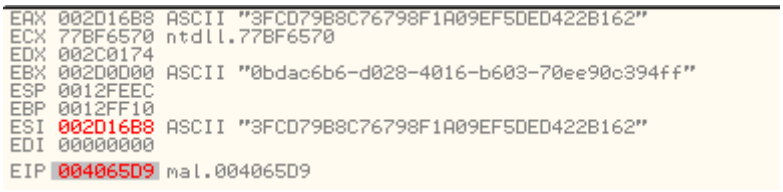




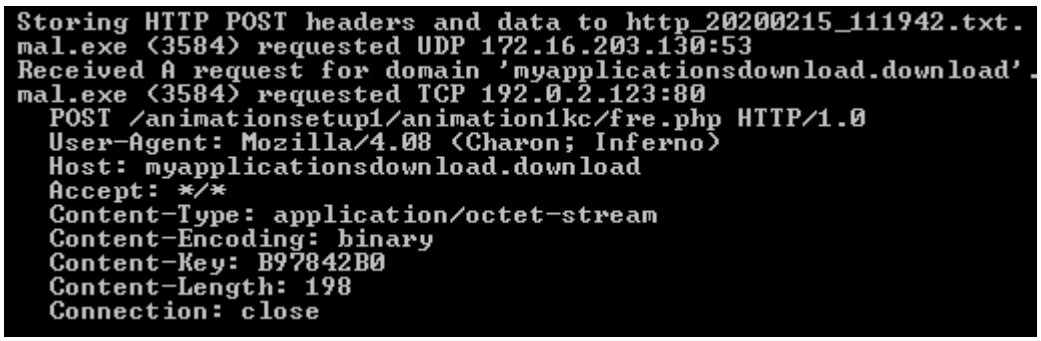
Malware set registry key and value but this record path is non-standart like Run and RunOnce paths used to achieve persistence.
Key:
HKU\S-1-5-21-3008613138-2701604480-1576304458-1001\myapplicationsdownload.download/animationsetup1/animation1kc/fre.php
Value:
/animation1kc/fre.php\8C7679: "%APPDATA%\8C7679\98F1A0.exe

Malware get MachineGuid registry value and calculate it's MD5. Generated MD5 value parsed to used as malware copy's directory and name.
MachineGuid : 0bdac6b6-d028-4016-b603-70ee90c394ff
MD5: 3FCD79B8C76798F1A09EF5DED422B162

Also mutex created that according MD5 hash.
Created Mutex: 3FCD79B8C76798F1A09EF5DE



To connect to C2 server malware uses WS2_32 getaddrinfo to resolve domain. I can not further network analysis because during analysis domain inactive.
Domain : myapplicationsdownload.download
Port : 80
Protocol : HTTP
User-Agent : Mozilla/4.08 (Charon; Inferno)



Dependencies

This analysis was performed on Windows 7 32-bit OS. Your test system must be connected to internet to running of malware main functionalities like communicate with C2, sending of stolen data.

Some DLLs which used by malware:
lsasrv.dll is an important security DLL which decrypts all local password hashing schemes on the computer.
vaultcli.dll, Vaultcmd.exe (and its dependency vaultcli.dll) are the command-line equivalent to the Credential Manager
URLs: http://myapplicationsdownload.download/animationsetup1/animation1kc/fre.php