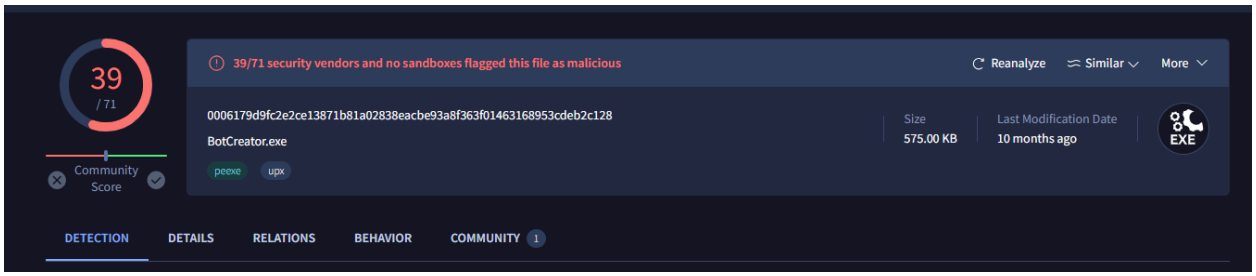
Brian McKeown
CSCE 451

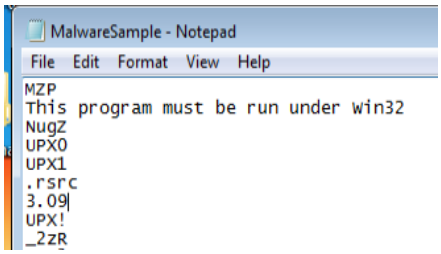
# Final Project Malware Analysis

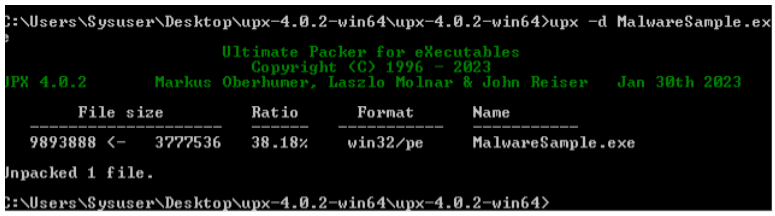
## Fingerprint:




## Unpacking the Malware:

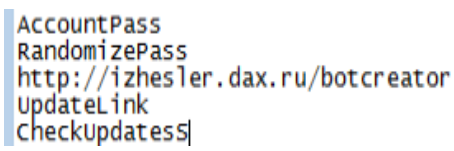
Upon downloading my malware sample, I ran strings and saw that it had been compressed with UPX. This meant I had to use “upx -d FinalMalwareSample.exe” to unpack the sample.

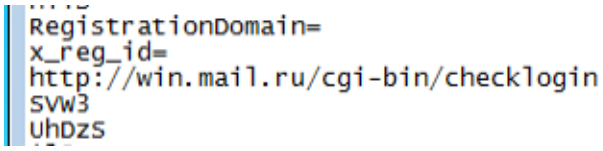


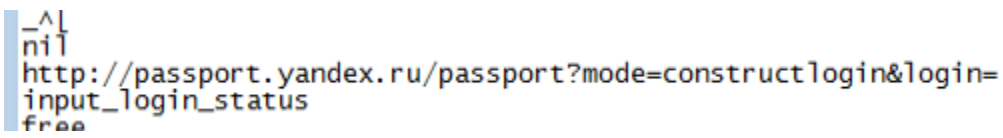




## Strings of Interest:

After unpacking, I look for anything that may be suspicious by running strings once again and find these suspicious URLs.



```
_^[
nil
http://id.rambler.ru/script/newuser.cgi
back=
step=one
login=
firstname=Onotole
lastname=Vasserman
```

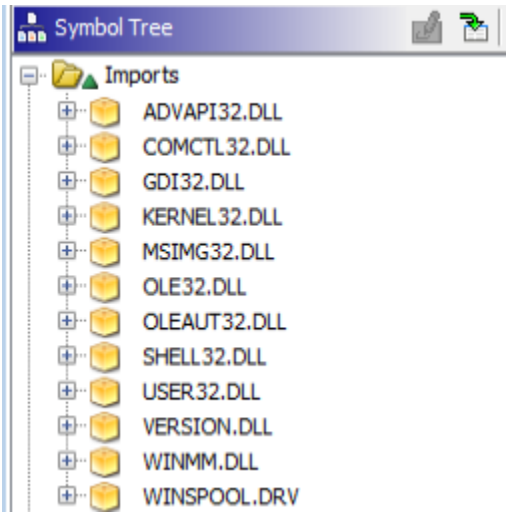
These were just four of the domains that stood out to me after scrolling through the txt file. I used VirusTotal's URL checker to find out that they are not inherently malicious but are in fact Russian.  Afterwards, I used PEStudio to flag strings it found particularly interesting.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ascii | 12 | section:.idata | x | import | memory | T1055 \| Process Injection | VirtualAlloc |
| ascii | 12 | section:.idata | x | import | memory | T1055 \| Process Injection | VirtualAlloc |
| ascii | 13 | section:.idata | x | import | input-output | T1056 \| Input Capture | MapVirtualKey |
| ascii | 16 | section:.idata | x | import | input-output | T1179 \| Hooking | GetKeyboardState |
| ascii | 11 | section:.idata | x | import | input-output | T1056 \| Input Capture | GetKeyState |
| ascii | 14 | section:.idata | x | import | input-output | - | GetKeyNameText |
| ascii | 15 | section:.idata | x | import | input-output | T1179 \| Hooking | GetKeyboardType |
| ascii | 19 | section:.idata | x | import | hooking | T1179 \| Hooking | UnhookWindowsHookEx |
| ascii | 16 | section:.idata | x | import | hooking | T1179 \| Hooking | SetWindowsHookEx |
| ascii | 14 | section:.idata | x | import | hooking | T1179 \| Hooking | CallNextHookEx |
| ascii | 9 | section:.idata | x | import | file | - | WriteFile |
| ascii | 12 | section:.idata | x | import | file | T1083 \| File and Directory Discovery | FindNextFile |
| ascii | 13 | section:.idata | x | import | file | T1083 \| File and Directory Discovery | FindFirstFile |
| ascii | 13 | section:.idata | x | import | file | T1083 \| File and Directory Discovery | FindFirstFile |
| ascii | 9 | section:.idata | x | import | file | - | WriteFile |

These are only a few of the flagged strings but ones that stood out to me were GetKeyState which makes me think of Keylogging and WriteFile which could have many malicious purposes.

**Imports:**

These are the imported libraries,




Some methods of interest include these 49 which were flagged by PEStudio.

| imports (475) | flag (49) | first-thunk-original (INT) | first-thunk (IAT) | hint | group (19) | technique (13) | type (5) | ordinal (1) | library (0) |
|---|---|---|---|---|---|---|---|---|---|
| GetWindowTextW | x | n/a | 0x0014C1FE | 0 (0x0000) | windowing | T1010 \| Window Discovery | implicit | - | user32.dll |
| GetWindowTextA | x | n/a | 0x0014C20E | 0 (0x0000) | windowing | T1010 \| Window Discovery | implicit | - | user32.dll |
| GetForegroundWindow | x | n/a | 0x0014C42C | 0 (0x0000) | windowing | T1010 \| Window Discovery | implicit | - | user32.dll |
| GetDesktopWindow | x | n/a | 0x0014C44C | 0 (0x0000) | windowing | - | implicit | - | user32.dll |
| GetClassLongA | x | n/a | 0x0014C4BC | 0 (0x0000) | windowing | - | implicit | - | user32.dll |
| EnumThreadWindows | x | n/a | 0x0014C548 | 0 (0x0000) | windowing | - | implicit | - | user32.dll |
| WinHelpA | x | n/a | 0x0014BD10 | 0 (0x0000) | shell | - | implicit | - | user32.dll |
| CopyImage | x | n/a | 0x0014C6E8 | 0 (0x0000) | resource | - | implicit | - | user32.dll |
| WritePrivateProfileStringA | x | n/a | 0x0014AA66 | 0 (0x0000) | registry | - | implicit | - | KERNEL32.DLL |
| RegFlushKey | x | n/a | 0x0014B37C | 0 (0x0000) | registry | T1112 \| Modify Registry | implicit | - | advapi32.dll |
| QueryPerformanceFrequency | x | n/a | 0x0014AB68 | 0 (0x0000) | reconnaissance | - | implicit | - | KERNEL32.DLL |
| GetCurrentProcessId | x | n/a | 0x0014AE4C | 0 (0x0000) | reconnaissance | T1057 \| Process Discovery | implicit | - | KERNEL32.DLL |
| VirtualAlloc | x | n/a | 0x0014AAB2 | 0 (0x0000) | memory | T1055 \| Process Injection | implicit | - | KERNEL32.DLL |
| VirtualAlloc | x | n/a | 0x0014B096 | 0 (0x0000) | memory | T1055 \| Process Injection | implicit | - | KERNEL32.DLL |
| MapVirtualKeyA | x | n/a | 0x0014C0AC | 0 (0x0000) | input-output | T1056 \| Input Capture | implicit | - | user32.dll |
| GetKeyboardState | x | n/a | 0x0014C3C0 | 0 (0x0000) | input-output | T1179 \| Hooking | implicit | - | user32.dll |
| GetKeyState | x | n/a | 0x0014C3FE | 0 (0x0000) | input-output | T1056 \| Input Capture | implicit | - | user32.dll |
| GetKeyNameTextA | x | n/a | 0x0014C40C | 0 (0x0000) | input-output | - | implicit | - | user32.dll |
| GetKeyboardType | x | n/a | 0x0014C7C2 | 0 (0x0000) | input-output | T1179 \| Hooking | implicit | - | user32.dll |
| UnhookWindowsHookEx | x | n/a | 0x0014BD48 | 0 (0x0000) | hooking | T1179 \| Hooking | implicit | - | user32.dll |
| SetWindowsHookExA | x | n/a | 0x0014BE0E | 0 (0x0000) | hooking | T1179 \| Hooking | implicit | - | user32.dll |
| CallNextHookEx | x | n/a | 0x0014C736 | 0 (0x0000) | hooking | T1179 \| Hooking | implicit | - | user32.dll |
| WriteFile | x | n/a | 0x0014AA82 | 0 (0x0000) | file | - | implicit | - | KERNEL32.DLL |
| FindNextFileA | x | n/a | 0x0014AF06 | 0 (0x0000) | file | T1083 \| File and Directory Discovery | implicit | - | KERNEL32.DLL |
| FindFirstFileA | x | n/a | 0x0014AF16 | 0 (0x0000) | file | T1083 \| File and Directory Discovery | implicit | - | KERNEL32.DLL |
| FindFirstFileA | x | n/a | 0x0014B254 | 0 (0x0000) | file | T1083 \| File and Directory Discovery | implicit | - | KERNEL32.DLL |
| WriteFile | x | n/a | 0x0014B298 | 0 (0x0000) | file | - | implicit | - | KERNEL32.DLL |
| SHGetSpecialFolderLocation | x | n/a | 0x0014BC80 | 0 (0x0000) | file | - | implicit | - | shell32.dll |
| SHGetFileInfoA | x | n/a | 0x0014BCCE | 0 (0x0000) | file | - | implicit | - | shell32.dll |
| GetCurrentThreadId | x | n/a | 0x0014AE38 | 0 (0x0000) | execution | T1057 \| Process Discovery | implicit | - | KERNEL32.DLL |
| GetCurrentThreadId | x | n/a | 0x0014B0F0 | 0 (0x0000) | execution | T1057 \| Process Discovery | implicit | - | KERNEL32.DLL |
| ShellExecuteA | x | n/a | 0x0014BCBE | 0 (0x0000) | execution | T1106 \| Execution through API | implicit | - | shell32.dll |
| GetWindowThreadProcessId | x | n/a | 0x0014C1CE | 0 (0x0000) | execution | T1057 \| Process Discovery | implicit | - | user32.dll |
| RaiseException | x | n/a | 0x0014AB58 | 0 (0x0000) | exception | - | implicit | - | KERNEL32.DLL |
| RaiseException | x | n/a | 0x0014B2F2 | 0 (0x0000) | exception | - | implicit | - | KERNEL32.DLL |
| GlobalFindAtomA | x | n/a | 0x0014AC68 | 0 (0x0000) | data-exchange | - | implicit | - | KERNEL32.DLL |
| GlobalDeleteAtom | x | n/a | 0x0014AC7A | 0 (0x0000) | data-exchange | - | implicit | - | KERNEL32.DLL |
| GlobalAddAtomA | x | n/a | 0x0014AC9A | 0 (0x0000) | data-exchange | - | implicit | - | KERNEL32.DLL |
| SetClipboardData | x | n/a | 0x0014BF04 | 0 (0x0000) | data-exchange | T1115 \| Clipboard Data | implicit | - | user32.dll |
| RegisterClipboardFormatA | x | n/a | 0x0014BFBE | 0 (0x0000) | data-exchange | T1115 \| Clipboard Data | implicit | - | user32.dll |
| OpenClipboard | x | n/a | 0x0014C02E | 0 (0x0000) | data-exchange | T1115 \| Clipboard Data | implicit | - | user32.dll |
| GetClipboardData | x | n/a | 0x0014C48A | 0 (0x0000) | data-exchange | T1115 \| Clipboard Data | implicit | - | user32.dll |

Some of the more familiar and dangerous functions I identify in this list include: GetKeyState, GetKeyboardState, SetWindowsHookExA, WriteFile, and MapVirtualKeyA. These are some of the methods that I will start my search with in the disassembled view in Ghidra.

**Disassembly:**

I started my analysis in Ghidra by checking and finding the main function. I did this by tracing to it from the entry point. This allowed me to understand and trace the execution of the program.

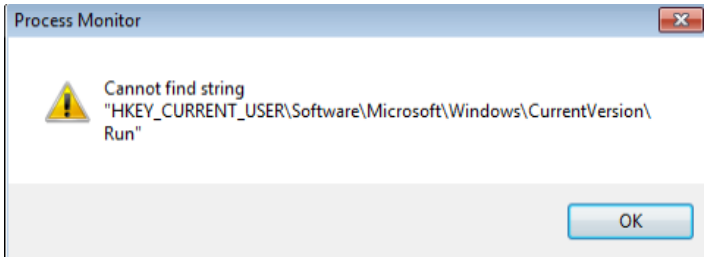
I then tried looking at the method GetKeyState and GetKeyboardState but they were both only referenced once in the code each at different unsuspicious locations. This made me look back at the imports where I found OLE32.dll. This was familiar from the CTFs that used COM interface. I checked references to the oleinitialize method and found FUN_0048e43c which calls FUN_0048e170 which retrieves and modifies all sorts of files in your system.

```
void FUN_0048e170(void)

{
  IShellFolder *in_FS_OFFSET;
  IShellFolder IVar1;
  IShellFolder IStack24;
  undefined *puStack20;
  undefined *puStack16;
  IShellFolder local_c;
  LPITEMIDLIST local_8;

  puStack16 = &stack0xfffffffc;
  local_c = (IShellFolder)0x0;
  puStack20 = &DAT_0048e1e1;
  IStack24 = *in_FS_OFFSET;
  *(IShellFolder **)in_FS_OFFSET = &IStack24;
  IVar1 = DAT_005455f0;
  SHGetSpecialFolderLocation((HWND)0x0,(int)DAT_005455f0,&local_8);
  if (local_8 != (LPITEMIDLIST)0x0) {
    FUN_0048e160((IShellFolder **)&local_c);
    IVar1 = local_c;
    DAT_00545654 = FUN_0048e208((int *)&PTR_FUN_0048dfb4,1,0,(int **)local_c,local_8);
    FUN_0048e110();
  }
  *in_FS_OFFSET = IVar1;
  IStack24 = (IShellFolder)&LAB_0048e1e8;
  FUN_0040646c((int **)&local_c);
  return;
}
```

```
4  void UndefinedFunction_0048e43c(void)
5
6  {
7    bool bVar1;
8
9    bVar1 = _DAT_00548d1c == 0;
10   _DAT_00548d1c = _DAT_00548d1c + -1;
11   if (bVar1) {
12     FUN_0048e170();
13     InitializeCriticalSection((LPCRITICAL_SECTION)&lpCriticalSection_00548d20);
14     OleInitialize((LPVOID)0x0);
15   }
16   return;
17 }
18
```

**Procmon:**

Here are some of the ways I have analyzed this malware for persistence mechanisms:




I defaulted to this as this is what was discussed in class and in the textbook but I could not find anything containing the path “\Software\Microsoft\Windows\CurrentVersion\Run”.

Next, I looked for any DLL injections. I did this by searching for “\SOFTWARE\Wow6432Node\Microsoft\Windows NT\CurrentVersion\Windows\AppInit_DLLs” but this was the closest thing I could find.


Additionally, it only queries that value so it does not seem too suspicious but I’m still documenting it in case it does become important later.

Next, I check to see if it potentially installs itself as a service so I make a note of these RegSetInfoKey operations in the “\SYSTEM\CurrentControlSet\Services” path.

| PID | Path | Time of Day | Relative Time | Operation | Result | Detail |
|---|---|---|---|---|---|---|
| 500 | HKLM\System\CurrentControlSet\services\LDAP | 9:07:44.1516328 ... | 00:18:42.6922873 | RegSetInfoKey | SUCCESS | KeySetInformation... |
| 500 | HKLM\System\CurrentControlSet\services\LDAP | 9:07:44.1516863 ... | 00:18:42.6923408 | RegSetInfoKey | SUCCESS | KeySetInformation... |
| 500 | HKLM\System\CurrentControlSet\services\LanmanWorkstation\Parameters | 9:07:44.2083054 ... | 00:18:42.7489599 | RegSetInfoKey | SUCCESS | KeySetInformation... |
| 500 | HKLM\System\CurrentControlSet\services\WinSock2\Parameters | 9:07:44.3318191 ... | 00:18:42.8724736 | RegSetInfoKey | SUCCESS | KeySetInformation... |
| 500 | HKLM\System\CurrentControlSet\services\WinSock2\Parameters | 9:07:44.3347820 ... | 00:18:42.8754365 | RegSetInfoKey | SUCCESS | KeySetInformation... |
| 500 | HKLM\System\CurrentControlSet\services\Winsock\Parameters | 9:08:13.0137237 ... | 00:19:11.5543782 | RegSetInfoKey | SUCCESS | KeySetInformation... |
| 500 | HKLM\System\CurrentControlSet\services\Tcpip\Parameters\Winsock | 9:08:13.0138087 ... | 00:19:11.5544632 | RegSetInfoKey | SUCCESS | KeySetInformation... |
| 500 | HKLM\System\CurrentControlSet\services\Winsock\Setup Migration\Providers | 9:08:13.0138817 ... | 00:19:11.5545362 | RegSetInfoKey | SUCCESS | KeySetInformation... |
| 500 | HKLM\System\CurrentControlSet\services\Tcpip\Parameters\Winsock | 9:08:13.0139690 ... | 00:19:11.5546235 | RegSetInfoKey | SUCCESS | KeySetInformation... |
| 500 | HKLM\System\CurrentControlSet\services\Tcpip\Parameters | 9:08:13.0165181 ... | 00:19:11.5571726 | RegSetInfoKey | SUCCESS | KeySetInformation... |
| 500 | HKLM\System\CurrentControlSet\services\Dnscache\Parameters | 9:08:13.0165693 ... | 00:19:11.5572238 | RegSetInfoKey | SUCCESS | KeySetInformation... |
| 500 | HKLM\System\CurrentControlSet\services\Tcpip\Parameters | 9:08:13.0167135 ... | 00:19:11.5573680 | RegSetInfoKey | SUCCESS | KeySetInformation... |
| 500 | HKLM\System\CurrentControlSet\services\Dnscache\Parameters | 9:08:13.0167520 ... | 00:19:11.5574065 | RegSetInfoKey | SUCCESS | KeySetInformation... |
| 500 | HKLM\System\CurrentControlSet\services\Tcpip\Parameters | 9:08:13.0170471 ... | 00:19:11.5577016 | RegSetInfoKey | SUCCESS | KeySetInformation... |
| 500 | HKLM\System\CurrentControlSet\services\Dnscache\Parameters | 9:08:13.0170879 ... | 00:19:11.5577424 | RegSetInfoKey | SUCCESS | KeySetInformation... |
| 500 | HKLM\System\CurrentControlSet\services\NetBT\Linkage | 9:08:13.0410609 ... | 00:19:11.5817154 | RegSetInfoKey | SUCCESS | KeySetInformation... |
| 500 | HKLM\System\CurrentControlSet\services\WinSock2\Parameters | 9:08:15.2904677 ... | 00:19:13.8311222 | RegSetInfoKey | SUCCESS | KeySetInformation... |
| 500 | HKLM\System\CurrentControlSet\services\NetBT\Linkage | 9:08:15.3452822 ... | 00:19:13.8859367 | RegSetInfoKey | SUCCESS | KeySetInformation... |
| 500 | HKLM\System\CurrentControlSet\services\NetBT\Linkage | 9:08:17.6318100 ... | 00:19:16.1724645 | RegSetInfoKey | SUCCESS | KeySetInformation... |
| 500 | HKLM\System\CurrentControlSet\services\NetBT\Linkage | 9:08:19.9266231 ... | 00:19:18.4672776 | RegSetInfoKey | SUCCESS | KeySetInformation... |
| 500 | HKLM\System\CurrentControlSet\services\NetBT\Linkage | 9:08:22.2091891 ... | 00:19:20.7498436 | RegSetInfoKey | SUCCESS | KeySetInformation... |
| 500 | HKLM\System\CurrentControlSet\services\NetBT\Linkage | 9:08:32.6728342 ... | 00:19:31.2134887 | RegSetInfoKey | SUCCESS | KeySetInformation... |

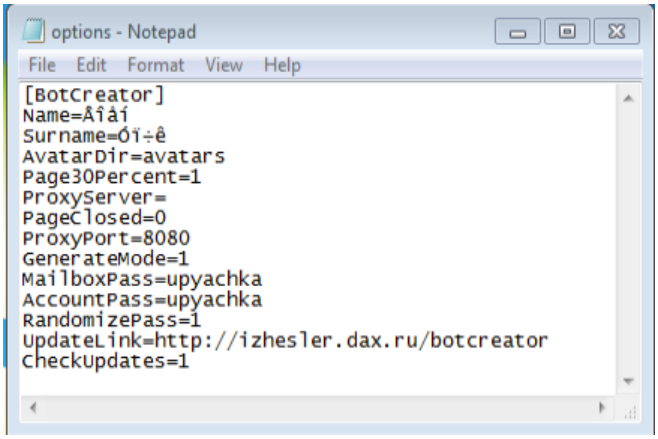
None of this appeared too out of the ordinary.

The following are the COM surrogates captured in the Procmon process tree when the malware sample was detonated.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| DllHost.exe (1908) | COM Surrogate | C:\Windows\syst... | Microsoft Corporat... | Sysuser-PC\Sysuser | C:\Windows\syst... | 4/30/2024 4:36:4... | 4/30/2024 4:36:4... |
| DllHost.exe (1276) | COM Surrogate | C:\Windows\syst... | Microsoft Corporat... | NT AUTHORITY\... | C:\Windows\syst... | 4/30/2024 4:36:5... | 4/30/2024 4:37:0... |
| DllHost.exe (1640) | COM Surrogate | C:\Windows\syst... | Microsoft Corporat... | NT AUTHORITY\... | C:\Windows\syst... | 4/30/2024 4:36:5... | 4/30/2024 4:37:0... |
| DllHost.exe (4072) | COM Surrogate | C:\Windows\syst... | Microsoft Corporat... | Sysuser-PC\Sysuser | C:\Windows\syst... | 4/30/2024 4:37:0... | 4/30/2024 4:37:1... |

The only file that the malware writes to is this options.ini file. I believe this may have to do with exfiltrating the data.

| PID | Path | Time of Day | Relative Time | Operation | Result | Detail |
|---|---|---|---|---|---|---|
| 3600 | C:\Users\Sysuser\Desktop\options.ini | 4:37:08.3494579 ... | 00:00:34.0938960 | WriteFile | SUCCESS | Offset: 19, Length: ... |
| 3600 | C:\Users\Sysuser\Desktop\options.ini | 4:37:08.3500855 ... | 00:00:34.0945236 | WriteFile | SUCCESS | Offset: 33, Length: ... |
| 3600 | C:\Users\Sysuser\Desktop\options.ini | 4:37:08.3505449 ... | 00:00:34.0949830 | WriteFile | SUCCESS | Offset: 49, Length: ... |
| 3600 | C:\Users\Sysuser\Desktop\options.ini | 4:37:08.3508901 ... | 00:00:34.0953282 | WriteFile | SUCCESS | Offset: 72, Length: ... |
| 3600 | C:\Users\Sysuser\Desktop\options.ini | 4:37:08.3515520 ... | 00:00:34.0959901 | WriteFile | SUCCESS | Offset: 100, Length... |
| 3600 | C:\Users\Sysuser\Desktop\options.ini | 4:37:08.3519813 ... | 00:00:34.0964194 | WriteFile | SUCCESS | Offset: 113, Length... |
| 3600 | C:\Users\Sysuser\Desktop\options.ini | 4:37:08.3523942 ... | 00:00:34.0968323 | WriteFile | SUCCESS | Offset: 132, Length... |
| 3600 | C:\Users\Sysuser\Desktop\options.ini | 4:37:08.3527726 ... | 00:00:34.0972107 | WriteFile | SUCCESS | Offset: 147, Length... |
| 3600 | C:\Users\Sysuser\Desktop\options.ini | 4:37:08.3531471 ... | 00:00:34.0975852 | WriteFile | SUCCESS | Offset: 169, Length... |
| 3600 | C:\Users\Sysuser\Desktop\options.ini | 4:37:08.3534976 ... | 00:00:34.0979357 | WriteFile | SUCCESS | Offset: 193, Length... |
| 3600 | C:\Users\Sysuser\Desktop\options.ini | 4:37:08.3539809 ... | 00:00:34.0984190 | WriteFile | SUCCESS | Offset: 207, Length... |
| 3600 | C:\Users\Sysuser\Desktop\options.ini | 4:37:08.3543265 ... | 00:00:34.0987646 | WriteFile | SUCCESS | Offset: 255, Length... |



## INetSim:

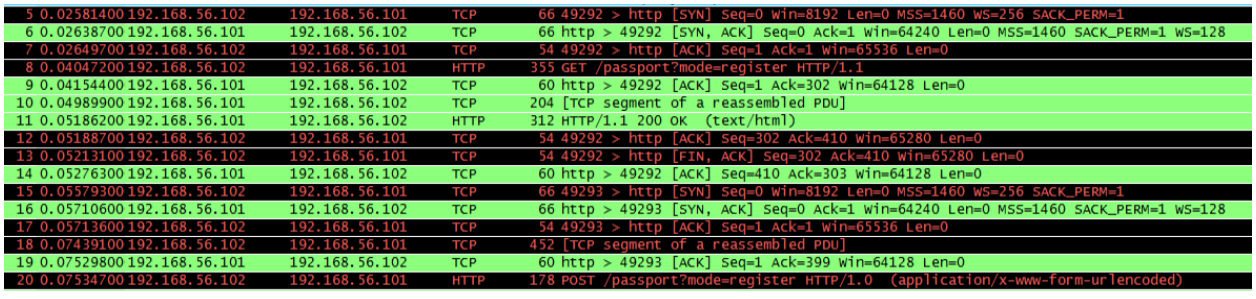
== Report for session '6335' ===

Real start date : 2024-04-30 11:09:38
Simulated start date : 2024-04-30 11:09:38
Time difference on startup : none

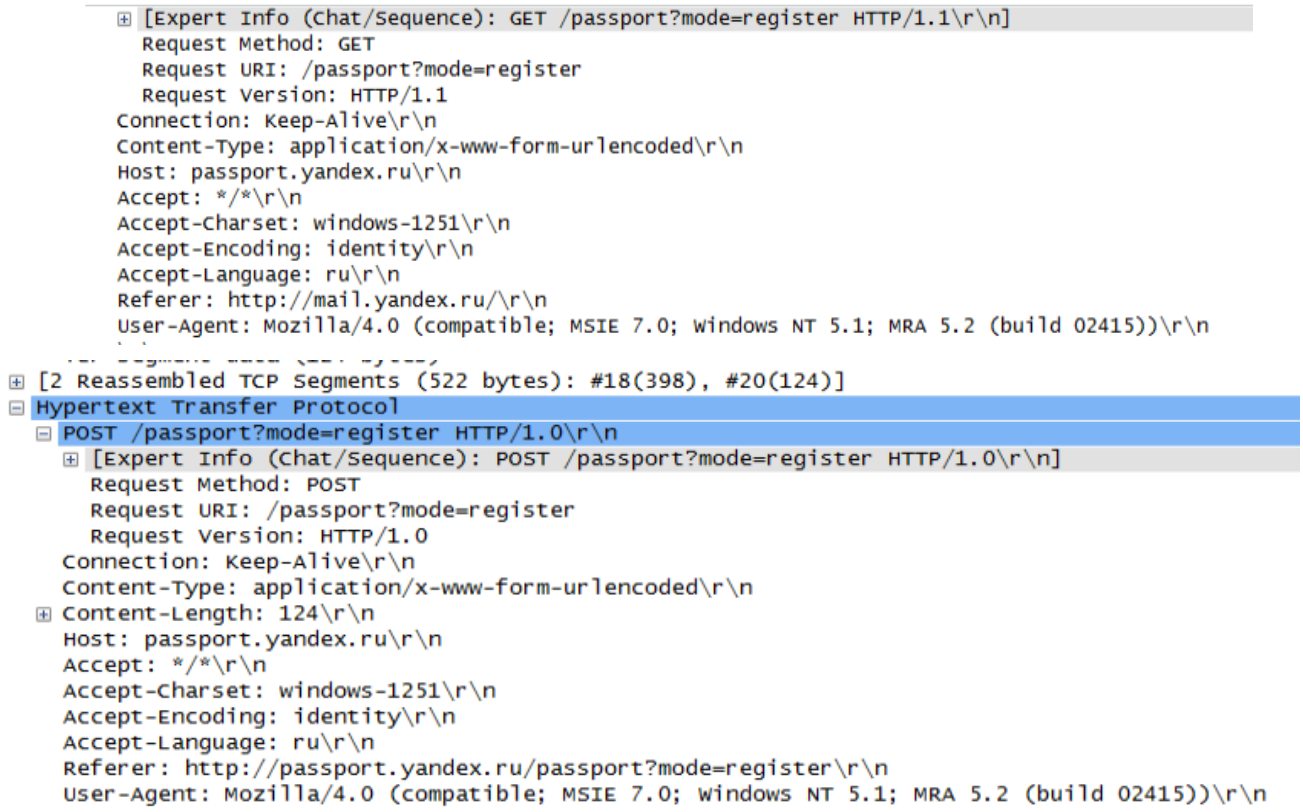
2024-04-30 11:09:42 First simulated date in log file
2024-04-30 11:09:42 DNS connection, type: A, class: IN, requested name: www.practicalmalwareanalysis.com
2024-04-30 11:09:42 HTTP connection, method: GET, URL: http://www.practicalmalwareanalysis.com/start.htm, file name: /var/lib/inetsim/http/fakefiles/sample.html
2024-04-30 11:09:58 DNS connection, type: A, class: IN, requested name: teredo.ipv6.microsoft.com
2024-04-30 11:10:59 DNS connection, type: A, class: IN, requested name: passport.yandex.ru
2024-04-30 11:10:59 HTTP connection, method: GET, URL: http://passport.yandex.ru/passport?mode=register, file name: /var/lib/inetsim/http/fakefiles/sample.html
2024-04-30 11:10:59 HTTP connection, method: POST, URL: http://passport.yandex.ru/passport?mode=register, file name: /var/lib/inetsim/http/postdata/c6d8150ebfb764d189350b12349063b3c9ecfe48285f4e6c18408f5d83a69>
2024-04-30 11:10:59 HTTP connection, method: GET, URL: http://passport.yandex.ru/digits?idkey=, file name: /var/lib/inetsim/http/fakefiles/sample.html
2024-04-30 11:11:04 HTTP connection, method: GET, URL: http://passport.yandex.ru/passport?mode=register, file name: /var/lib/inetsim/http/fakefiles/sample.html
2024-04-30 11:11:04 HTTP connection, method: POST, URL: http://passport.yandex.ru/passport?mode=register, file name: /var/lib/inetsim/http/postdata/c6d8150ebfb764d189350b12349063b3c9ecfe48285f4e6c18408f5d83a69>
2024-04-30 11:11:04 HTTP connection, method: GET, URL: http://passport.yandex.ru/digits?idkey=, file name: /var/lib/inetsim/http/fakefiles/sample.html
2024-04-30 11:11:14 DNS connection, type: A, class: IN, requested name: watson.microsoft.com
2024-04-30 11:11:15 HTTP connection, method: GET, URL: http://watson.microsoft.com/StageOne/FinalMalwareSample_exe/0_2_1_0/2a425e19/FinalMalwareSample_exe/0_2_1_0/2a425e19/c0000005/0006980a.htm?LCID=1033&OS=6.>
2024-04-30 11:11:15 Last simulated date in log file

===

## Wireshark:



| | | | | | |
|---|---|---|---|---|---|
| 5 | 0.02581400 | 192.168.56.102 | 192.168.56.101 | TCP | 66 49292 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 6 | 0.02638700 | 192.168.56.101 | 192.168.56.102 | TCP | 66 http > 49292 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128 |
| 7 | 0.02649700 | 192.168.56.102 | 192.168.56.101 | TCP | 54 49292 > http [ACK] Seq=1 Ack=1 Win=65536 Len=0 |
| 8 | 0.04047200 | 192.168.56.102 | 192.168.56.101 | HTTP | 355 GET /passport?mode=register HTTP/1.1 |
| 9 | 0.04154400 | 192.168.56.101 | 192.168.56.102 | TCP | 60 http > 49292 [ACK] Seq=1 Ack=302 Win=64128 Len=0 |
| 10 | 0.04989900 | 192.168.56.101 | 192.168.56.102 | TCP | 204 [TCP segment of a reassembled PDU] |
| 11 | 0.05186200 | 192.168.56.101 | 192.168.56.102 | HTTP | 312 HTTP/1.1 200 OK (text/html) |
| 12 | 0.05188700 | 192.168.56.102 | 192.168.56.101 | TCP | 54 49292 > http [ACK] Seq=302 Ack=410 Win=65280 Len=0 |
| 13 | 0.05213100 | 192.168.56.102 | 192.168.56.101 | TCP | 54 49292 > http [FIN, ACK] Seq=302 Ack=410 Win=65280 Len=0 |
| 14 | 0.05276300 | 192.168.56.101 | 192.168.56.102 | TCP | 60 http > 49292 [ACK] Seq=410 Ack=303 Win=64128 Len=0 |
| 15 | 0.05579300 | 192.168.56.102 | 192.168.56.101 | TCP | 66 49293 > http [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 |
| 16 | 0.05710600 | 192.168.56.101 | 192.168.56.102 | TCP | 66 http > 49293 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM=1 WS=128 |
| 17 | 0.05713600 | 192.168.56.102 | 192.168.56.101 | TCP | 54 49293 > http [ACK] Seq=1 Ack=1 Win=65536 Len=0 |
| 18 | 0.07439100 | 192.168.56.102 | 192.168.56.101 | TCP | 452 [TCP segment of a reassembled PDU] |
| 19 | 0.07529800 | 192.168.56.101 | 192.168.56.102 | TCP | 60 http > 49293 [ACK] Seq=1 Ack=399 Win=64128 Len=0 |
| 20 | 0.07534700 | 192.168.56.102 | 192.168.56.101 | HTTP | 178 POST /passport?mode=register HTTP/1.0 (application/x-www-form-urlencoded) |




From the activity in Inetsim and Wireshark, I am led to believe that this malware uses network connection to exfiltrate the data that it captures. I believe it may potentially use the Russian yandex email service to send the data to some unknown attacker as it is mentioned in the HTTP GET request.

**Ollydbg:**

Nothing that was not discovered previously in the static and dynamic analysis were found in the debugger. However, I did learn that if you close the program while it is running it crashes trying to access memory out of bounds to write to.


**How to Clean Machine:**

1. Disconnect the machine from the internet and any other networks to prevent the malware from receiving any further instructions from a remote server.
2. Restart the machine in safe mode. This prevents any non-essential programs from running.

3. Backup important files that you do not want to lose. Make sure not to back up any .exe files as they may be infected.
4. Use Antivirus and Anti-Malware Programs to scan and remove malicious programs.
5. Clear system restore points as these can still hold copies of the malware.
6. Reset all browsers
7. Update your software
8. Change all your passwords
9. Monitor your systems behavior going forward to make sure all steps taken did in fact clean your system.

**Conclusion:**

This malware is a sophisticated multi-faceted approach that tries to obfuscate its intentions by employing an unsuspecting GUI on run name botcreator.exe. I can't tell exactly what the "purpose" of the GUI is because it is in russian but it does not matter. Upon running the executable, the malware spreads through the file system and uses the COM interface to collect data and modify registries. Lastly, the malware sends the data through network connection to Russian URLs.