PseudoManuscrypt Being Distributed in the Same Method as Cryptbot

asec.ahnlab.com/en/31683/

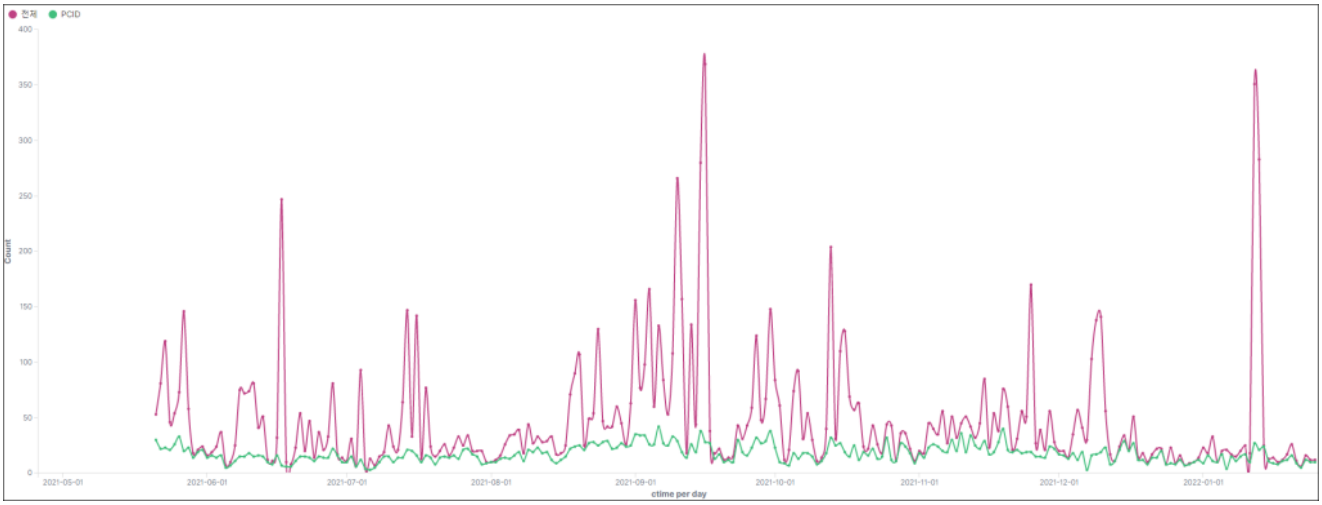
February 18, 2022

The ASEC analysis team has discovered that PseudoManuscrypt malware was being distributed in Korea since May 2021. Introduced in the previous ASEC blog, PseudoManuscrypt is disguised as an installer that is similar to a form of Cryptbot, and is being distributed. Not only is its file form similar to Cryptbot, but it is also distributed via malicious sites exposed on the top search page when users search commercial software-related illegal programs such as Crack and Keygen.

The team has confirmed the executable file path below in the logs collected by AhnLab's ASD (AhnLab Smart Defense) infrastructure, and it appears that the user was trying to download a Windows validation program from a malicious site.

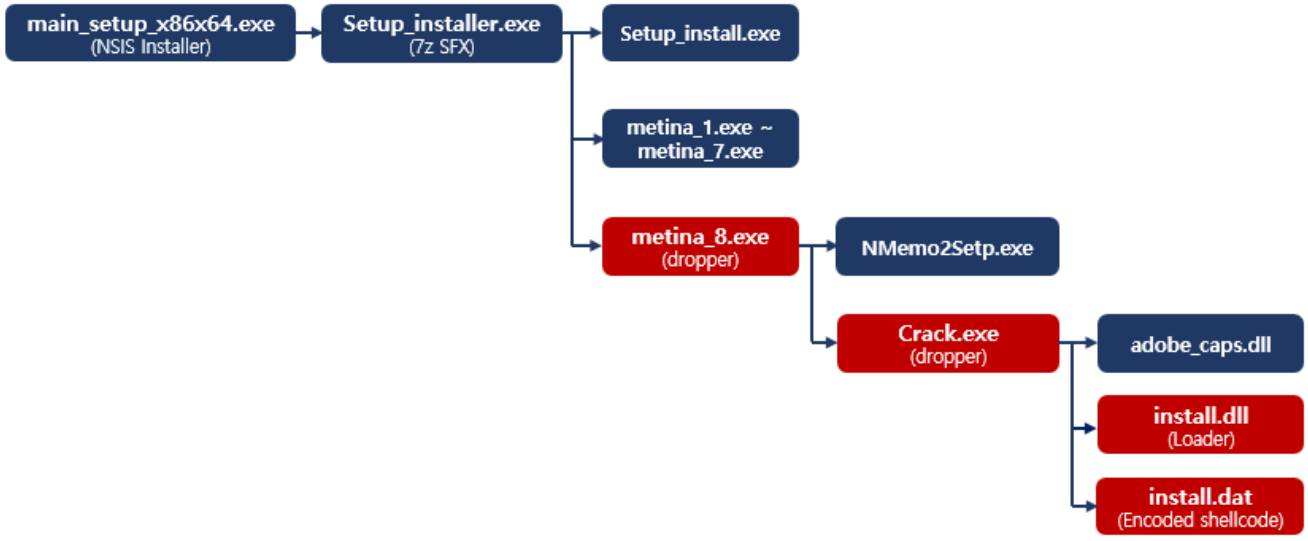
Executable File Path

…\downloads\60b63e_kmsauto-net-201\kmsauto-net-2016-v154-windows-10-activator-portable\60b63e21e82a660b63e21e_setup_v18.2.9\main_setup_x86x64.exe



Log Detection Graph

Such a distribution method targets random users, and it has been confirmed that numerous PCs in Korea were infected. The figure above is a graph of the number of logs that were detected since the start of distribution (May 2021) up till now. The green graph shows the number of infected PCs, and the red graph shows the number of detected files. You can see that on average, around 30 PCs were consistently being infected every day.

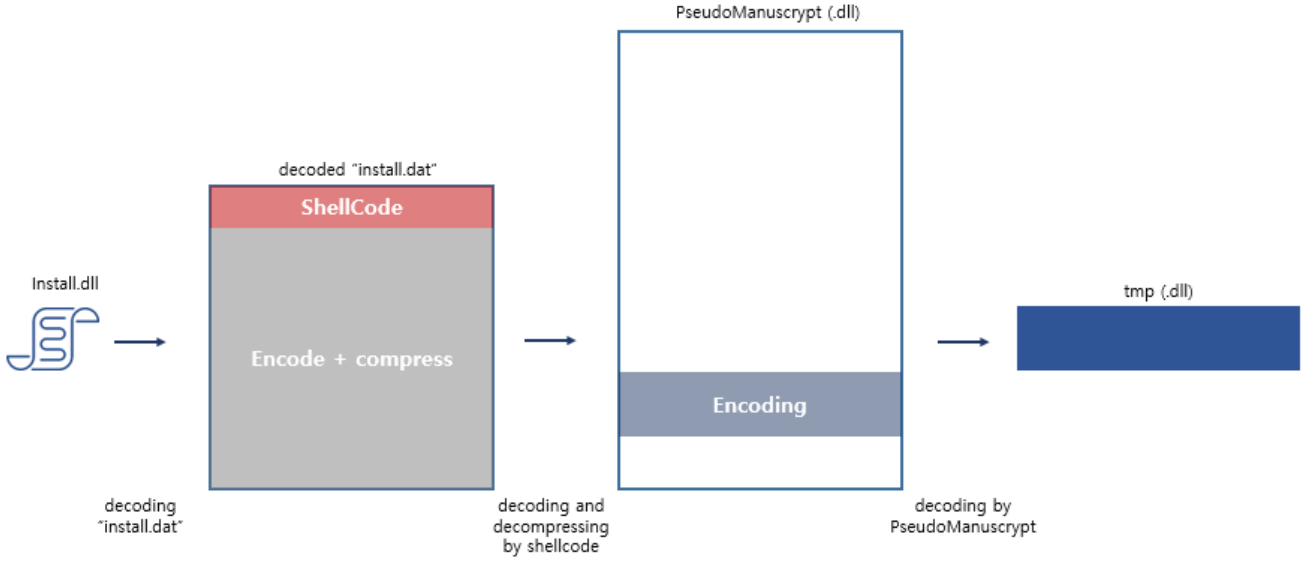


PseudoManuscrypt operation flow

The top-level file disguised as an illegal program is in the form of NSIS (Nullsoft Scriptable Install System) Installer, and it creates the “setup_installer.exe” file to execute it. “Setup_installer.exe” is in the form of 7z SFX, and it creates a Loader file, various malware, and numerous dll files. The dll files are all normal files, and they are libraries needed to execute the Loader file. Loader executes the various malware that was created together, and this process is the same as the execution process of Cryptbot. Other than PseudoManuscrypt, malware executed by Loader includes SmokeLoader, Glupteba, etc., and PseudoManuscrypt is in the form of 7z SFX. Finally, PseudoManuscrypt creates install.dll (Loader that performs decoding) and install.dat (Encoded shellcode) in the %TEMP% path. It then creates and executes a shortcut file called “install.dll.lnk” to operate a certain function inside the install.dll file.

Property of install.dll.lnk

C:\Windows\system32\rUNdlL32.eXe “%TEMP%\install.dll”,install



PseudoManuscrypt execution flow (2)

The install.dll file decodes the install.dat file to execute it in the memory. The install.dat file contains a shellcode and pe data that is encoded and compressed, and the shellcode decodes the encoded and compressed pe data and executes it. The pe data then performs the actual malicious behavior and decodes additional data inside to create it with the tmp extension, which is then registered to service. Details about pe data are explained below.

```
v0 = GetCommandLineW();
GetModuleFileNameW(0, &Filename, 0x104u);
v1 = wcsrchr(&Filename, 0x5Cu);
if ( lstrcmpiW(v1 + 1, L"svchost.exe") )
{
  if ( sub_669580() )
  {
    result = (PWSTR)sub_66A700(v0);
  }
  else
  {
    v8 = CreateThread(0, 0, sub_669C40, 0, 0, &ThreadId);
    WaitForSingleObject(v8, 0xFFFFFFFF);
    result = (PWSTR)CloseHandle(v8);
  }
}
else if ( StrStrIW(v0, L"netsvcs") )
{
  v2 = CreateThread(0, 0, sub_6694C0, 0, 0, 0);
  CloseHandle(v2);
  v3 = CreateThread(0, 0, sub_6689C0, 0, 0, &ThreadId);
  result = (PWSTR)CloseHandle(v3);
}
else if ( StrStrIW(v0, L"SystemNetworkService") )
{
  v5 = CreateThread(0, 0, sub_669C40, 0, 0, &ThreadId);
  WaitForSingleObject(v5, 0xFFFFFFFF);
  result = (PWSTR)CloseHandle(v5);
}
else
{
  result = StrStrIW(v0, L"AppService");
```

Command line scan

The called pe data first checks if the name of the currently running process is svchost.exe. In this case, the current process was run as rundll32.exe by the shortcut file.

MachineGuid value

When it is not a svchost.exe process, it creates a certain registry key and saves malicious data. For the name of the created registry, it uses the MachineGuid value that exists in the HKLM\SOFTWARE\Microsfot\Cryptography registry. MachineGuid is a HardwareID value, where each PC has a unique value. It uses this characteristic to create the registry key name.

MachineGuid value is encoded via the “Global” string, and the encoded MachineGuid value is used to create a registry key in the path below. The encoded “install.dat” data is then saved to the registry key.

– HKLM\SOFTWARE\Classes\CLSID\MachineGuid(“Global”):1 – Encoded “install.dat”

```
if ( EnumServicesStatusExW(
       hSCObject,
       SC_ENUM_PROCESS_INFO,
       0x30u,
       1u,
       (LPBYTE)v5,
       v18,
       &pcbBytesNeeded,
       &ServicesReturned,
       &ResumeHandle,
       0) )
{
  v7 = (void (__stdcall *)(SC_HANDLE))CloseServiceHandle;
  v8 = (SC_HANDLE)LocalAlloc(0x40u, 0x2000u);
  v9 = 0;
  hSCObject = v8;
  v22 = 0;
  v25 = 0;
  if ( ServicesReturned )
  {
    v10 = (DWORD *)((char *)hMem + 36); // process id
    do
    {
      if ( *(v10 - 6) == 4 )            // 실행 중인 서비스인지
      {
        v11 = OpenServiceW(v6, (LPCWSTR)*(v10 - 9), 1u);// v10-9 : service name
        if ( v11 )
        {
          if ( QueryServiceConfigW(v11, (LPQUERY_SERVICE_CONFIGW)hSCObject, 0x2000u, &v22) )
          {
            v12 = (const WCHAR *)*((_DWORD *)hSCObject + 3);// path
            if ( v12 )
            {
              if ( StrStrIW(v12, L"-k netsvcs") )
              {
                v13 = *v10;
                for ( i = 0; i < 0x64; ++i )
                {
                  v15 = dwProcessId[i];
```

Running service scan

After it creates the registry key, it checks the process id of services that include “-k netsvcs” in their arguments, then injects the decoded “install.dat” in the process. It then deletes the “install.dat” file that exists in the %TEMP% path, and the injected process references the encoded “install.dat” data that was saved to the registry key.

The injected svchost.exe goes through the same process and scans the command line. It checks if the name of the current process is svchost.exe and if netsvcs is included in the argument, it executes two threads.

The threat that is first executed performs the feature of registering a certain file to service when the process is terminated. It first brings down the priority of process termination to the lowest and configures the control handler.

```
BOOL __stdcall HandlerRoutine(DWORD CtrlType)
{
  if ( CtrlType == 5 )
  {
    SetProcessShutdownParameters(0, 0);
    SetConsoleCtrlHandler(HandlerRoutine, 1);
  }
  else if ( CtrlType == 6 )
  {
    sub_669410();
    return 0;
  }
  return 0;
```

Control handler function

The function that is added to the control handler is as shown above, and it checks the received control signals. The function scans for the following control signal values: 5 and 6. If the signal value is 5 (When the user logs off), it reconfigures the priority of process termination and the control handler, and if the signal value is 6 (When the system shuts down), it executes a function that creates a malicious service. The malicious service is created via the process below.

After creating the SYSTEM\\CurrentControlSet\\Services\\AppService[a-z] registry, it configures this service registry as shown below.

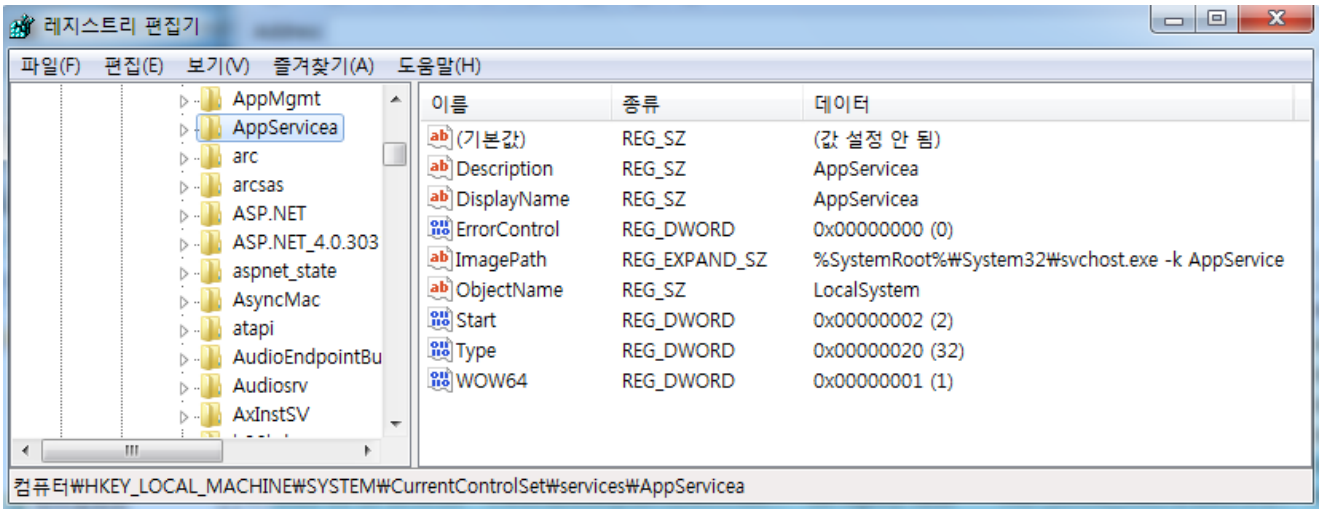
– Start : 0x02 (Starts automatically when the system starts)

– imagepath : %SystemRoot%\System32\svchostexe -k AppService

The name of the created service and the file that is executed via the service are configured as follows:

– SYSTEM\\CurrentControlSet\\ServicesAppService[a-z]\\Parameters:servicedll = %System%\Encoded string.tmp

– SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Svchost:AppService =AppService[a-z]

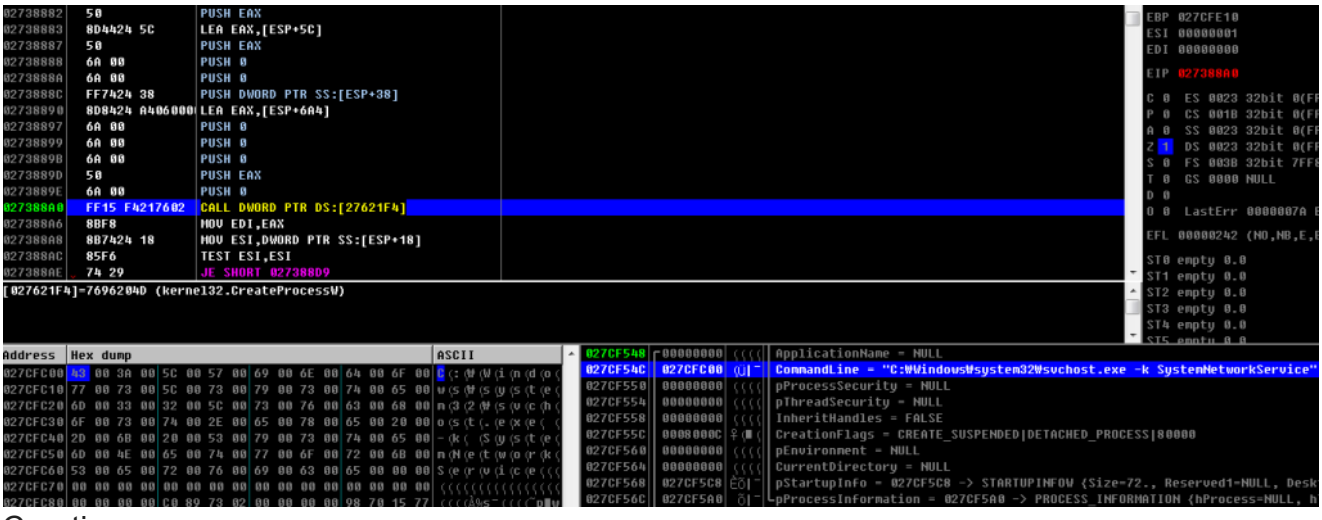


Created service

The file that is to be registered to the service decodes the data that exists in a certain location and creates it with the tmp extension in the System path. The created tmp file acts as a Loader that decodes and executes the encoded “install.dat” data existing in a certain registry key. It appears that this process serves the purpose of performing continuous malicious behaviors even when the user PC is shut down and restarted.

Adding an exclusion to Windows Defender

After the service is created, the System folder is excluded from Windows Defender scans, and the first thread is terminated.



Creating process

The second thread that is executed is the %System%\svchost.exe -k SystemNetworkService command, which performs the function of creating a process and injecting the decoded “install.dat” data.

The injected svchost.exe goes through the same process to scan the command line, and if it’s svchost.exe run by the SystemNetworkService argument, it performs the actual malicious behavior. It steals various user credentials including the data below via this process and sends them to the attacker’s server.

- VPN connection information
- Clipboard data
- Audio data
- List of shared network folders
- Information of processes that accept TCP and UDP ports
- File version information of the running process
- C2 : email.yg9[.]me

In addition to the malicious features above, it can also access the C2 server under the attacker’s command and perform various malicious behaviors such as file download, screen capture, and execution of keylogger and cmd commands.

As this malware is disguised as an illegal software installer and is distributed to random individuals via malicious sites, users must be careful not to download relevant programs. As malicious files can also be registered to service and perform continuous malicious behaviors without the user knowing, periodic PC maintenance is necessary.

[File Detection]

- Trojan/Win.Generic.R420870 (2021.05.16.01)
- Malware/Win.Generic.R421780 (2021.05.21.03)
- Trojan/Win.Generic.C4512227 (2021.06.04.01)
- Trojan/Win.Generic.C4512246 (2021.06.04.01)
- Trojan/Win.Generic.R421722 (2021.08.17.03)
- Trojan/Win.Generic.R436809 (2021.08.17.03)
- Trojan/Win.Generic.R436811 (2021.08.17.03)
- Trojan/Bin.Encoded (2022.01.28.02)

[IOC Info]

- 1fecb6eb98e8ee72bb5f006dd79c6f2f
- 5de2818ced29a1fedb9b24c1044ebd45
- 58efaf6fa04a8d7201ab19170785ce85
- 839e9e4d6289eba53e40916283f73ca6
- 89c8e5a1e24f05ede53b1cab721c53d8
- 5e6df381ce1c9102799350b7033e41df

- a29e7bbe6dee4eea95afa3f2e3a1705a
- 8ae40c8418b2c36b58d2a43153544ddd
- email.yg9[.]me

Subscribe to AhnLab's next-generation threat intelligence platform 'AhnLab TIP' to check related IOC and detailed analysis information.

Categories:Malware Information

Tagged as:PseudoManuscrypt