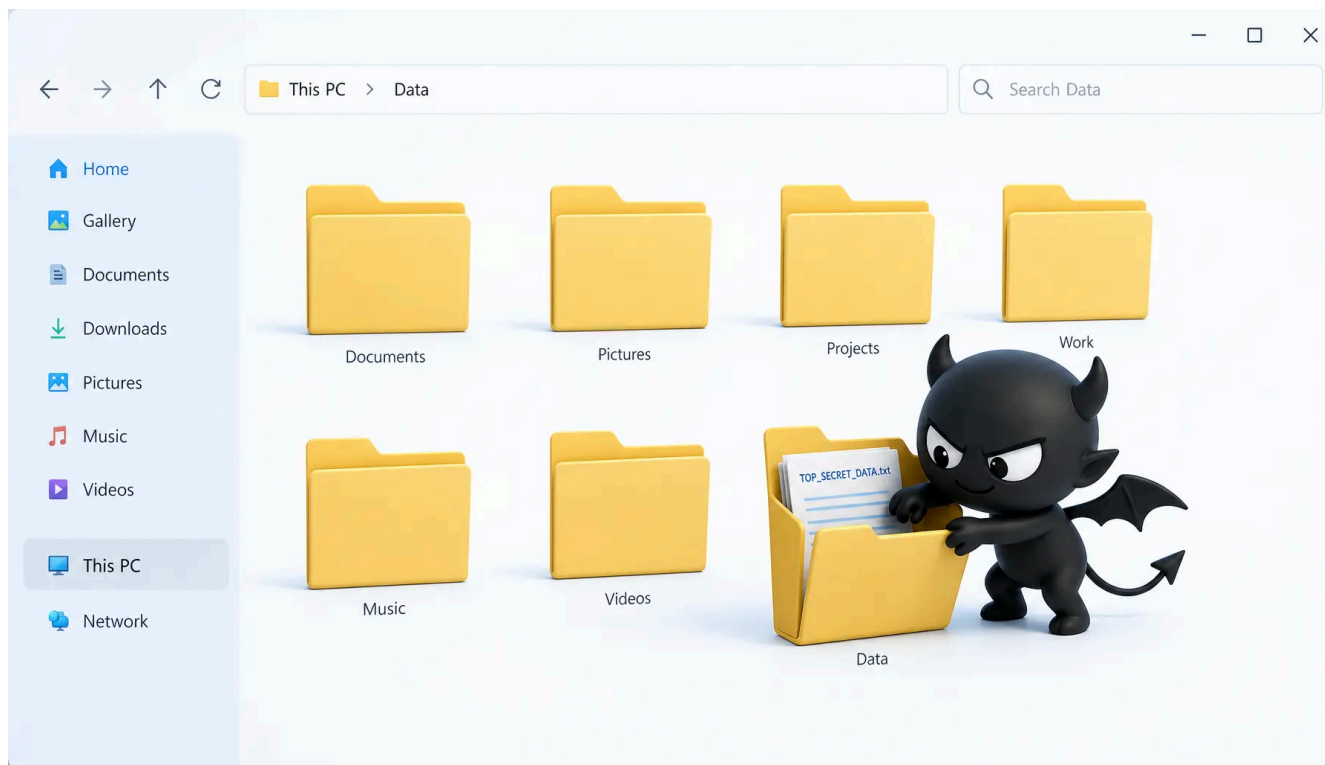


Beware of phishing emails disguised as requests to review quotes (PhantomStealer)

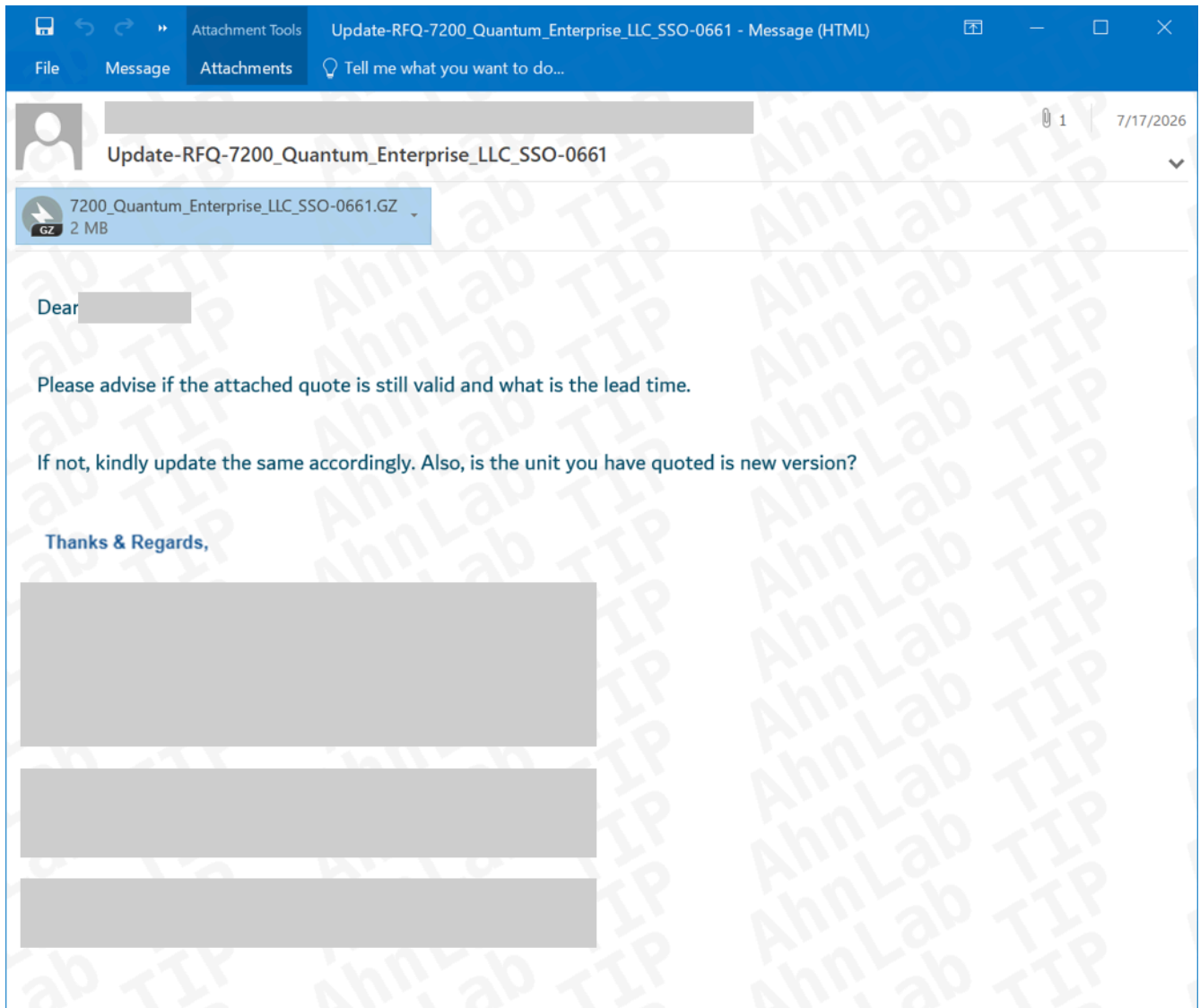
asec.ahnlab.com/en/95000

ATCP

August 11, 2026

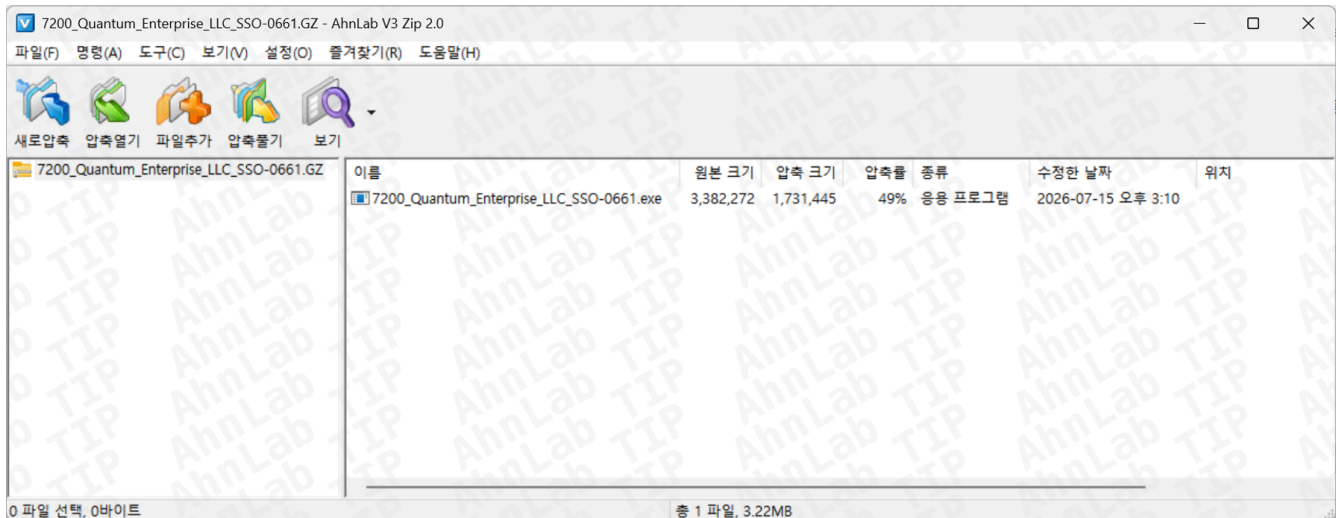


The AhnLab Security intelligence Center (ASEC) recently identified a phishing email campaign that disguised itself as a request to review a quote. The threat actor impersonated a sales team member at a specific overseas company and, by claiming that a previous quote needed to be revised and product versions verified, tricked recipients into opening the Attachment. The email contained a malicious file named “7200_Quantum_Enterprise_LLC_SSO-0661.GZ” , which contained a malicious compressed file.



[Figure 1] Phishing email body

Inside the attachment is a file named “7200_Quantum_Enterprise_LLC_SSO-0661.Exe.” This file is an “injector”-type malware that exploits vulnerable drivers to disable security software and then injects and executes the PhantomStealer malware into legitimate processes.



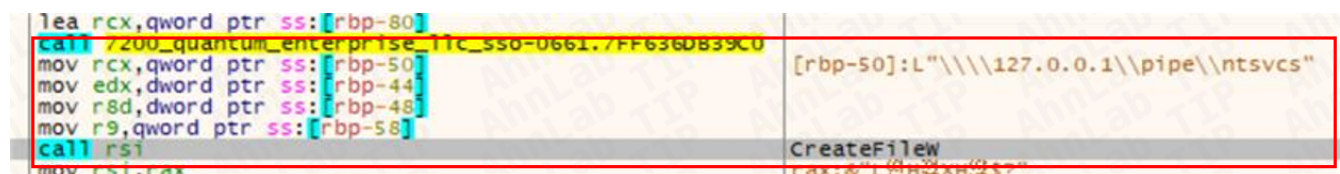
[Figure 2] Compressed file

1. UAC Privilege Escalation

The malware first checks whether the current process is running with administrator privileges. If it already possesses an administrator token, it skips the UAC bypass procedure and proceeds to carry out subsequent malicious actions. Conversely, if it is running with standard user privileges, it sequentially attempts SSPI-based privilege escalation and automatic CMSTPLUA COM elevation; if automatic elevation fails, it uses the `runas` behavior of `ShellExecuteExW` as a last resort.

1.1. SSPI-Based UAC Bypass

In an environment where the malware belongs to the Administrators group but is running at a medium integrity level due to UAC, it performs NTLM- or Negotiate-based SSPI Datagram authentication on its own to obtain a network authentication token. It then connects to `\\127.0.0.1\Pipe\ntsvcs` while impersonating the acquired token, prompting the loopback authentication process to use the elevated token. It then calls SCM RPC through the acquired pipe to create and start a Service that runs the current sample. Since no separate service account is specified during this process, the sample ultimately runs with LocalSystem privileges.



[Figure 3] Code that uses CreateFileW to open the `\\127.0.0.1\Pipe\ntsvcs` named pipe

It then calls the CreateServiceW function to register the created DCRCVDrv.Sys file as a kernel driver service with the Service Control Manager (SCM). The service name is NvStreamKmd_dcrvc; the driver is loaded into the kernel through the service start, preparing the system to call the driver's functions from user mode.

```

00007FF92CD7DFC3 mov qword ptr ss:[rsp+C0],rax
00007FF92CD7DFCF mov rax,qword ptr ds:[r11-38],rax
00007FF92CD7DFD3 mov rax,qword ptr ss:[rsp+88]
00007FF92CD7DFD8 mov qword ptr ds:[r11-40],rax
00007FF92CD7DFDF mov eax,dword ptr ss:[rsp+80],rax
00007FF92CD7DFE6 mov dword ptr ss:[rsp+30],eax
00007FF92CD7DFEA mov eax,dword ptr ss:[rsp+A8],eax
00007FF92CD7DFF1 mov dword ptr ss:[rsp+28],eax
00007FF92CD7DFF5 mov eax,dword ptr ss:[rsp+A0],eax
00007FF92CD7DFF6 mov dword ptr ss:[rsp+20],eax
00007FF92CD7E000 call qword ptr ds:[&CreateServiceW]
00007FF92CD7E007 nop dword ptr ds:[rax+rax],eax

```

[Figure 6] Registration of the NvStreamKmd_dcrvc kernel driver Service

```

C:\Users\...Desktop>sc queryex NvStreamKmd_dcrvc

SERVICE_NAME: NvStreamKmd_dcrvc
        종류               : 1  KERNEL_DRIVER
        상태               : 4  RUNNING
                               (STOPPABLE, NOT_PAUSABLE, IGNORES_SHUTDOWN)
        WIN32_EXIT_CODE      : 0  (0x0)
        SERVICE_EXIT_CODE  : 0  (0x0)
        검사점              : 0x0
        WAIT_HINT           : 0x0
        PID                 : 0
        플래그              :

```

[Figure 7] Registered NvStreamKmd_dcrvc Service

Once the driver is loaded, it accesses the \\.\DCRCVDRV_U device object through the CreateFileW function. This is not the behavior of opening a regular file, but rather the process of acquiring a handle to the communication channel between user mode and kernel mode provided by DCRCVDrv.Sys.

```

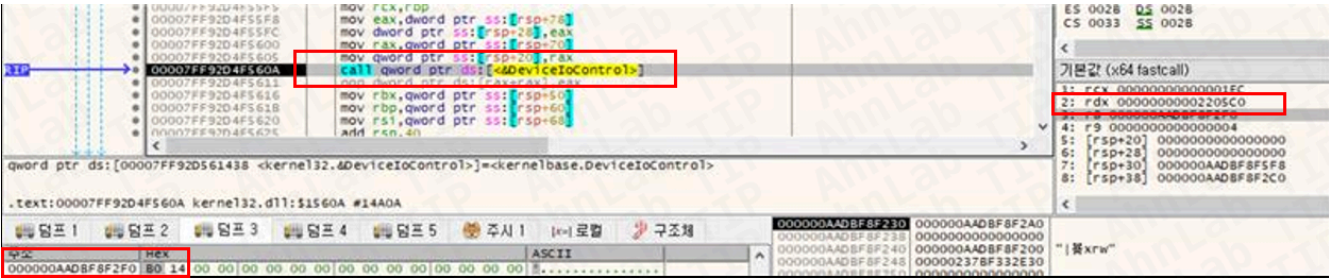
00007FF636EAD011 mov qword ptr ss:[rbp-58],r9
00007FF636EAD015 lea rcx,qword ptr ss:[rbp-80]
00007FF636EAD019 call 7200_quantum_enterprise_11c_sso-0661.7F
00007FF636EAD01E mov rcx,qword ptr ss:[rbp-50]
00007FF636EAD022 mov edx,dword ptr ss:[rbp-44]
00007FF636EAD025 mov r8d,dword ptr ss:[rbp-48]
00007FF636EAD029 mov r9,qword ptr ss:[rbp-58]
00007FF636EAD02D call rsi

```

[Figure 8] Acquiring the \\.\DCRCVDRV_U device object handle

The DeviceIoControl API is called on the acquired device handle to send a request to terminate the target process. The code 0x2205C0 used here is a driver-specific IOCTL code in DCRCVDrv.Sys that signifies a process termination request, and the PID of the process to be terminated is passed in the input buffer. Upon receiving the request, the driver acquires a process handle for the specified PID and then terminates the process by calling

ZwTerminateProcess in kernel mode. Through this method, the injector terminates security products—whose termination is restricted to user-mode privileges—using kernel-mode privileges.



[Figure 9] Request to terminate a target process through DeviceIoControl

The PID (5296) of SecurityHealthService.Exe—one of the security products targeted for termination—is passed as an input value to the DeviceIoControl API. It was confirmed that when the DeviceIoControl API is subsequently called, DCRCVDrv.Sys processes the request and terminates the SecurityHealthService.Exe process. The program names of the security products targeted for termination are listed in [Table 1].



[Figure 10] SecurityHealthService.Exe before executing the DeviceIoControl API



[Figure 11] SecurityHealthService.Exe terminated after the DeviceIoControl API execution

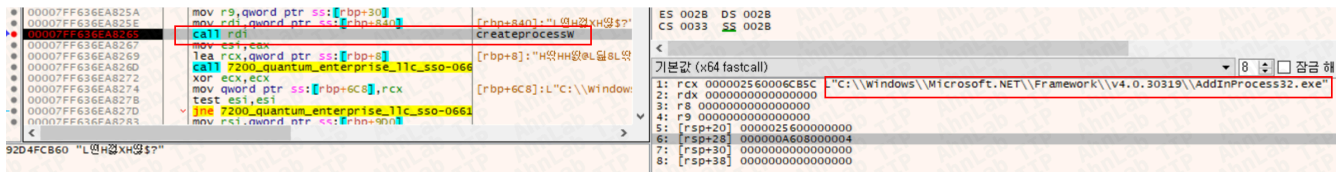
Notepad.Exe, CSFalconService.Exe, CSSensorSettings.Exe, MsMpEng.Exe, MsSense.Exe, MsSenseS.Exe, smartscreen.Exe, SecurityHealthService.Exe, SecurityHealthTray.Exe, SentinelAgent.Exe, SentinelCtl.Exe, cb.Exe, CyveraService.Exe, cytool.Exe, sfc.Exe, iptraytool.Exe, ipsupporttool.Exe, McShield.Exe, mfevtps.Exe, mfemms.Exe, mcagent.Exe, masvc.Exe, macmnsvc.Exe, mcods.Exe, McUICnt.Exe, ccSvcHst.Exe, Smc.Exe, SmcGui.Exe, SymCorpUI.Exe, SavService.Exe, SophosFS.Exe, SophosFileScanner.Exe, SSPService.Exe, SEDService.Exe, SophosAgent.Exe, SophosUI.Exe, hmpalert.Exe, TmListen.Exe, NTRtScan.Exe, TmPfw.Exe, TMBMSRV.Exe, TmCCSF.Exe, TmProxy.Exe, PccNTMon.Exe, avp.Exe, avpui.Exe, ekrn.Exe, egui.Exe, emsw.Exe, esets_psi.Exe, esets_main.Exe, bdagent.Exe, bdservicehost.Exe, vsserv.Exe, bdwtxag.Exe, bdntwrk.Exe, seccenter.Exe, AvastSvc.Exe, AvastUI.Exe, aswidsagent.Exe, aswToolsSvc.Exe, afwServ.Exe, wsc_proxy.Exe, aswEngSrv.Exe, avgsvc.Exe, avgwdsvc.Exe, MBAMService.Exe, mbam.Exe, mbamtray.Exe, WRSA.Exe, fctservctl2.Exe, fctsched.Exe, FortiEDRCollectorService.Exe, elastic-endpoint.Exe, fshoster32.Exe, PSANHost.Exe, PSUAService.Exe, CylanceSvc.Exe, CylanceUI.Exe, CybereasonService.Exe, CybereasonLauncher.Exe, CybereasonRansomFreeServiceHost.Exe, xagt.Exe, avcenter.Exe, avguard.Exe, sched.Exe, avshadow.Exe, webrootsecureanywhere.Exe, wrusrvsvc.Exe, wrusrv.Exe, panda_url_filtering.Exe, pavsvrx86.Exe, aviramain.Exe, avscan.Exe, avira.Servicehost.Exe, trgui.Exe, coreServicesShell.Exe, sophosav.Exe, surfshark.Exe, surfsharkservice.Exe, clamd.Exe, freshclam.Exe, drweb32w.Exe, drwt32.Exe, vipreui.Exe, sbamsvc.Exe, fprottray.Exe, gdatasecuritycenter.Exe, K7TSecurity.Exe, K7TSMngr.Exe, K7Sysmon.Exe, K7CrvSvc.Exe, K7AVScan.Exe, K7AVMScn.Exe, K7RTScan.Exe, K7CTScan.Exe, K7AVQrnt.Exe, K7FWSrv.Exe, K7NDFHlpr.Exe, K7EmIPxy.Exe, K7PSSrv.Exe, K7SpmSrc.Exe, K7APHlpr.Exe, K7TLActiveHsty.Exe, K7DisinfectorGUI.Exe, K7MebezatEncRemovalTool.Exe, K7QuervarCleaningTool.Exe, K7TSMMain.Exe, 360Tray.Exe, QHSafeMain.Exe, QHActiveDefense.Exe, QHWatchdog.Exe, QHSafeTray.Exe, 360rp.Exe, 360rps.Exe, 360Safe.Exe, zhudongfangyu.Exe, 360AdvToolExecutor.Exe, HipsDaemon.Exe, HipsMain.Exe, HipsTray.Exe, HRUpdate.Exe, HipsLog.Exe, HRConfig.Exe

[Table 1] List of security product program names to be terminated

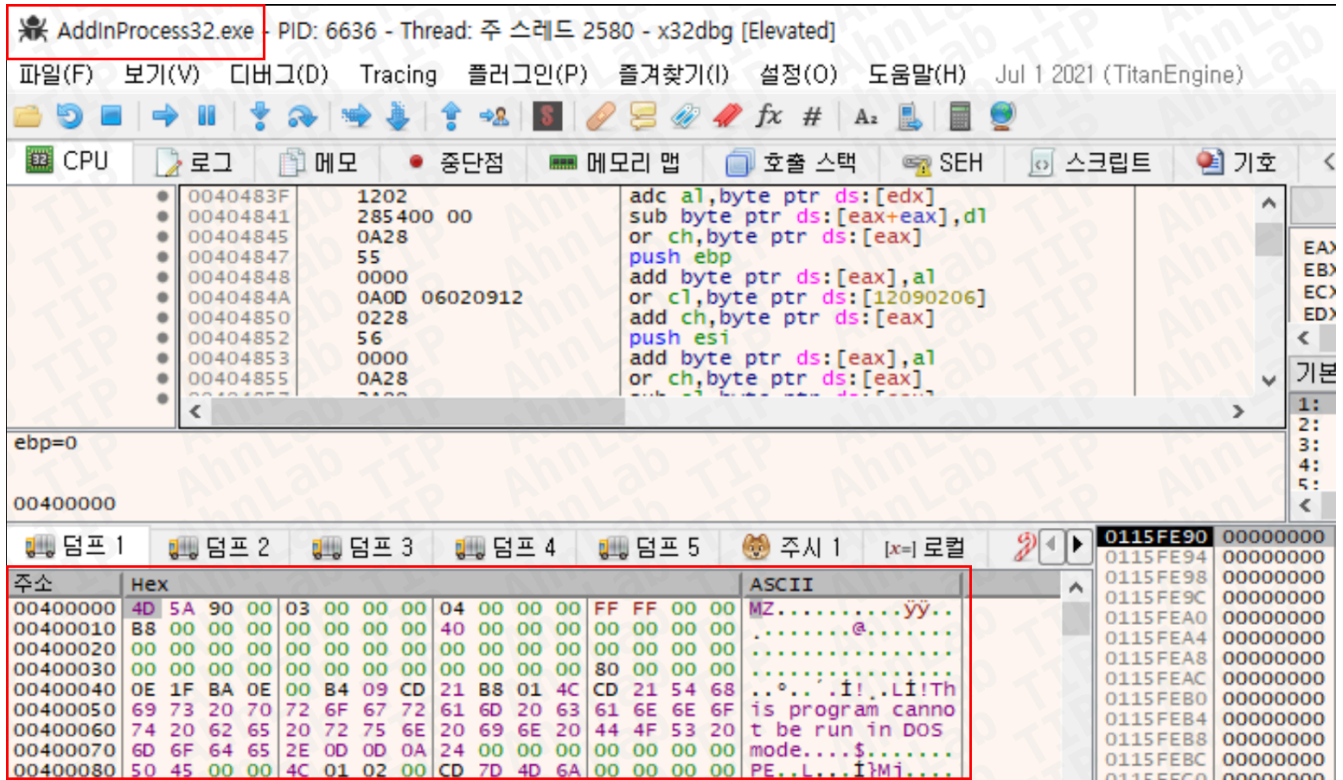
3. Process Hollowing

The injector performs process hollowing on the legitimate process

C:\Windows\Microsoft.NET\Framework\v4.0.30319\AddInProcess32.Exe. It then allocates a memory region in the target process through `VirtualAllocEx`, inserts the PhantomStealer payload through `WriteProcessMemory`, and calls `ResumeThread` to resume execution.



[Figure 12] Injection of the PhantomStealer payload into the AddInProcess32.Exe process



[Figure 13] PhantomStealer injected into the AddInProcess32.Exe process

4. PhantomStealer

The injected PhantomStealer runs within the context of the AddInProcess32.Exe process and collects data such as keystrokes and screen captures, as well as account information stored in browsers and applications, login session cookies, and system and network information. It also steals cryptocurrency wallet data and clipboard contents, and performs a “clipper” function that replaces cryptocurrency wallet addresses stored in the clipboard with the threat actor’s address.

C2

- Mail Server: mail.Mnt[.]Com:587
- Username: phan@trimnt.Com

```

string text2 = SystemInfo.GetSystemVersion();
string text3 = SystemInfo.Username;
string text4 = SystemInfo.Compname;
string text5 = Flags.GetFlag(SystemInfo.Culture.Split(new char[] { '-' })[1]);
string text6 = SystemInfo.Culture;
string text7 = SystemInfo.GetAntivirus();
string text8 = SystemInfo.GetDefaultGateway();
string text9 = SystemInfo.GetLocalIp();
string text10 = await SystemInfo.GetPublicIpAsync();
string text11 = string.Concat(new string[]
{
    "`\n\ud83d\ude39 *Phantom Stealer ",
    text,
    " - Credit Card Report:*~\n\n\n\ud83d\udcc5 Date: ",
    SystemInfo.Datetime,
    "\n\ud83d\udca5\ufe0f System: ",
    text2,
    "\n\ud83d\udc64 Username: ",
    text3,
    "\n\ud83d\udcbb CompName: ",
    text4,
    "\n\ud83c\udf10 Language: ",
    text5,
    " ",
    text6,
    "\n\ud83d\udee1\ufe0f Antivirus: ",
    text7,
    "\n\ud83c\udf10 Gateway IP: ",
    text8,
    "\n\ud83d\udd12 Internal IP: ",
    text9,
    "\n\ud83c\udf0d External IP: ",
    text10,
    "\n`"
});

```

[Figure 14] PhantomStealer's Information Theft capabilities

Information Theft and Related Items	Targets for Theft and Tampering
Keypresses	Keyboard characters and input content entered by the user
Screen Captures	Screen images displayed on the primary monitor
Stored Account Information	Account names, passwords, and host names stored in Chromium- and Gecko-based browsers, as well as in FileZilla, FoxMail, WinSCP, and Outlook
System and Network Information	Operating system, username, computer name, locale, antivirus software, default gateway, internal IP, and external IP
Cookies	Login session cookies stored in the browser
Credit card information	Credit card information stored in the browser or elsewhere
File Collection	Files and documents with specified file extensions or located in specified paths
Browser wallet information	Virtual asset wallet data based on browser extensions
Desktop wallet information	Data from virtual asset wallet applications installed on the system
Clipboard history	Text copied by the user to the clipboard
Clipper	Replaces virtual asset wallet addresses stored in the clipboard with the threat actor's address

[Table 2] PhantomStealer Features

Response Guide

Immediate Actions

- Attachment 7200QuantumEnterpriseLLCSS0-0661.GZ and the 7200QuantumEnterpriseLLCSS0-0661.Exe inside them should be treated as malicious files.

- Check whether the attachment was executed on the device that received the email; if it has been executed, classify the device as a suspected infection.
- Check for signs that security product processes have been terminated. The targets listed in the report are SecurityHealthService.Exe the names of numerous security products, including [Name].
- Check whether C:\Windows\Temp\DCRCVDrv.Sys Check whether a file has been created on the system.
- NvStreamKmd_dcrcv Check whether the service is registered.
- \\.\\DCRCVDRV_U Check for traces of access to device objects and DeviceIoControl call traces.
- Check for traces of IOCTL codes 0x2205C0 used to terminate the target process.
- AddInProcess32.Exe Check for process hollowing traces
- Assess the potential for data exfiltration regarding keystrokes, screen captures, browser account information, login session cookies, system and network information, virtual asset wallet data, and clipboard contents—all of which are targets of Information Theft by PhantomStealer.

MD5

0c14e6b5a3a8c1cf695b38a71adf8647
 567c158ee0858f8e941d4ab7a6c18dbc
 c155a21bec649260089a4a55c0afca43