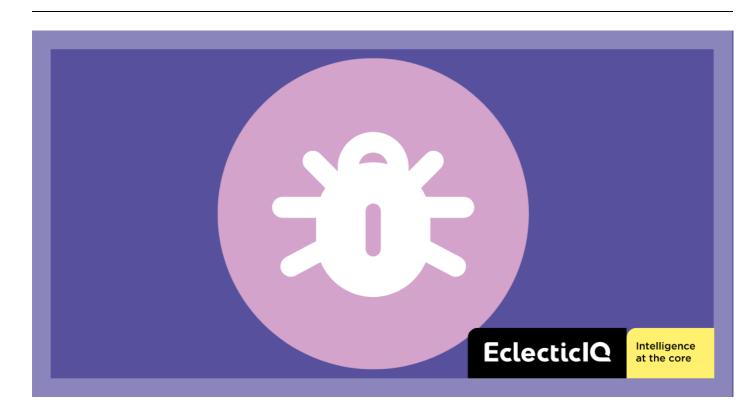
blog.eclecticiq.com /mustang-panda-apt-group-uses-european-commission-themed-lure-to-deliver-plugx-malware

Mustang Panda APT Group Uses European Commission-Themed Lure to Deliver PlugX Malware



EXECUTIVE SUMMARY

- Since at least 2019, the Mustang Panda threat actor group has targeted government and public sector organizations across Asia and Europe [3] with long-term cyberespionage campaigns in line with strategic interests of the Chinese government.
- In November 2022, Mustang Panda shifted from using archive files to using malicious optical disc image (ISO) files containing a shortcut (LNK) file to deliver the modified version of PlugX malware. This switch increases the evasion against anti-malware solutions [2].
- The Mustang Panda APT group loads the PlugX malware in the memory of legitimate software by employing a four-stage infection chain which leverages malicious shortcut (LNK) files, triggering execution via dynamic-link library (DLL) search-order-hijacking.

PLUGX MALWARE EXECUTION FLOW

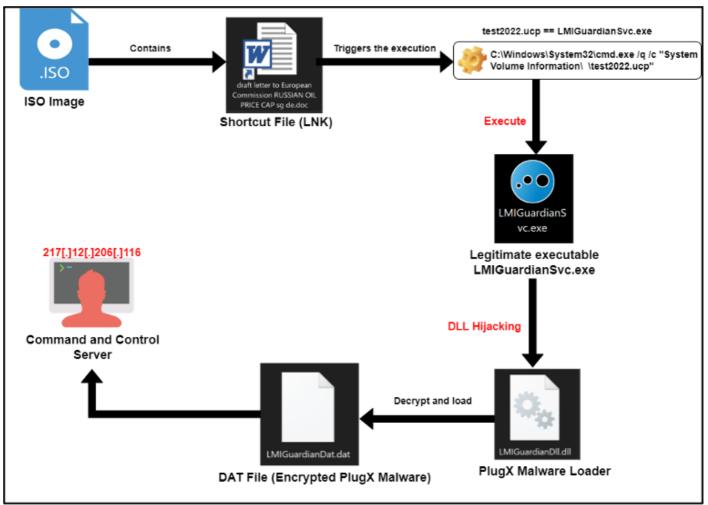


Figure 1 – Execution flow of PlugX malware.

First Stage: PlugX Malware Delivered by ISO Image

In the first stage of the infection chain, EclecticIQ researchers assess that the malware was almost certainly delivered by a malicious email with an ISO image attachment. The ISO image contains a shortcut (LNK) file, but it decoyed as a DOC file called "draft letter to European Commission RUSSIAN OIL PRICE CAP sg de.doc".

The malicious LNK file contains a command line argument that can be executed by user execution to start the PlugX malware execution chain.

The command line argument of "draft letter to European Commission RUSSIAN OIL PRICE CAP sg de.doc" is shown below:

C:\Windows\System32\cmd.exe /q /c "System Volume Information\ \test2022.ucp"

The test2022.ucp portion of the command line argument is a renamed legitimate software which is originally called LMIGuardianSvc.exe. This executable is abused to perform DLL hijacking and to load the initial PlugX loader called LMIGuardianDII.dll. The legitimate and malicious executables are placed on the same file path (System Volume Information) to perform DLL Hijacking.

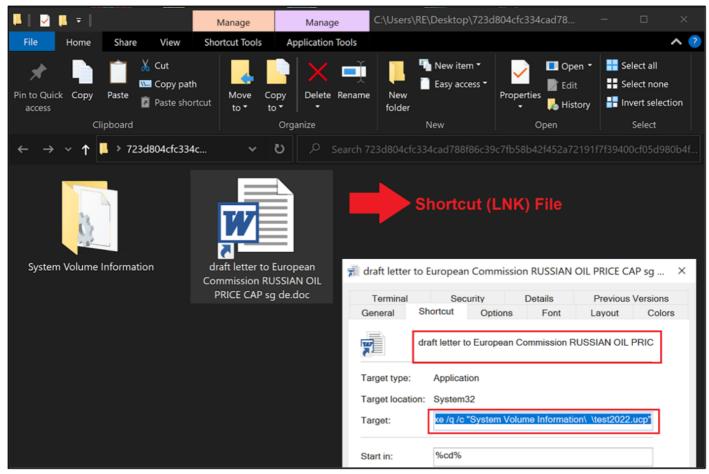


Figure 2 – Command line argument of malicious shortcut (LNK) file.

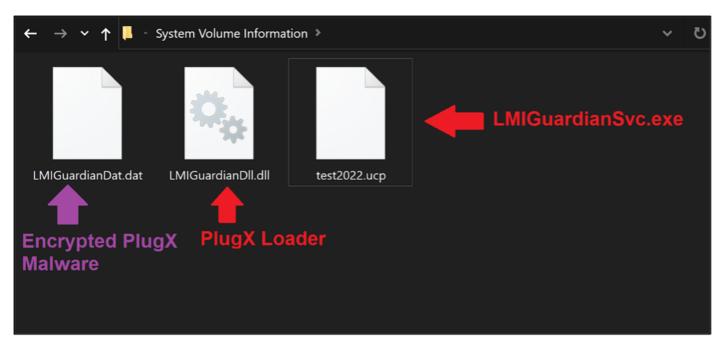


Figure 3 -PlugX malware loader execution file path.

Second Stage: DLL Hijacking Execution Chain to Load PlugX Malware

When a victim clicks on the shortcut file, it executes the command line argument mentioned in first stage, which is a technique called DLL hijacking (after the execution of LMIGuardianSvc.exe, it loads LMIGuardianDII.dll aka PlugX loader automatically). Upon execution of the PlugX loader a Microsoft Office Word document opens. The document is named "draft letter to European Commission RUSSIAN OIL

PRICE CAP sg de.docx". This is a decoy document to trick the user into thinking there is no malicious activity.

One example of the Word document can be seen in the image below:

∎ 5 0 ·	draft letter to European Commission. RUSSIAN OIL PRICE CAP sg de.docx - Word (Ürün Etkinleştirilemedi)
Dosya <mark>Giriş</mark> Ekle Tasarım Sayfa Düzeni Başvurular Posta	lar Gözden Geçir Görünüm 🛛 Ne yapmak istediğinizi söyleyin
Kes Calibri (Gövde) 12 A A = - A Yapistir Kigin Boyacisi Kigin Boyacisi A A A Pano Yazi Tipi Yazi Tipi	E · E · · · · · · · · · · · · · · · · ·
Pano in Yazi lipi i	raragrat · Stiller
	Draft letter: Cyprus, Greece and Malta CY 21.11.2022
	21 November 2022
	Re: Shipping related EU Restrictive Measures against the Russian Federation- Russian Oil Price Cap 1 From February 2022 to date, there has been a significant number of deletions of ships from the
	ship registries of Cyprus, Greece and Malta, which represent a significant percentage of the EU merchant fleet. Ships are being reflagged to non-EU States which do not implement the EU sanctions against the Russian Federation whilst it is not expected that <u>shipowners</u> will select to register any ships under EU ship registries.
	2 This will undoubtedly lead to the:
	 (a) loss of national revenue for EU Member States in the public and private sectors, (b) loss of a substantial share of the global fleet which is controlled by the EU, (c) weakening of the voice and influence of the EU at the relevant international fora of the UN (e.g. IMO-ILO- UNCTAD) and the negotiating position of the EU Member States in our join efforts to promote EU policies at global level (e.g. sustainable development and <u>decarbonisation</u> of shipping).

Figure 4 – A decoy Word document is used for social engineering. The victim sees a real Word document open after clicking on a shortcut (LNK) file that has a Word document icon.

The process tree below shows the execution of the legitimate application LMIGuardianSvc.exe, which is executed twice under a new directory (\AppData\Roaming\SamsungDriver) created by the malware and used for persistence access on infected device.

E cmd	.exe (1832)	Windows Command Pro	C:\Windows\System32\cmd.exe
Co	nhost.exe (6616)	Console Window Host	C:\Windows\System32\Conhost.exe
E test	2022.ucp (732)	LMIGuardianSvc	C:\Users\RE\Desktop\723d804cfc334cad788f86c39c7fb58b42f452a721
W	NWORD.EXE (3812)	Microsoft Word	C:\Program Files\Microsoft Office\Office16\WINWORD.EXE
LN	IIGuardianSvc.exe (9556)	LMIGuardianSvc	C:\Users\RE\AppData\Roaming\SamsungDriver\LMIGuardianSvc.exe
		<	
Description:	Windows Command Processor	test2022	2.ucp == LMIGuardianSvc.exe
Company:	Microsoft Corporation		
Path:	C:\Windows\System32\cmd.exe		
Command:	"C:\Windows\System32\cmd.exe	/q /c "System Volume Information\	\test2022.ucp"
User:	DESKTOP-5C29KH5\RE		

Figure 5 – Captured process tree during the execution of malicious shortcut (LNK) file which masquerades as a word document.

Encrypted shellcode named LMIGuardianDat.dat contains PlugX malware:

📓 LMIGuardia	nDat	.dat															
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	OF	Decoded text
00000000	DC	D3	D9	Fl	Α4	75	9E	C7	FO	41	24	E8	58	BD	1A	42	ÜÓÙñ¤užÇðA\$èX⅓.B
00000010	7D	34	FC	B5	A3	01	48	96	42	E9	2E	07	AD	A 8	8E	94	}4üµ£.H-B锎″
00000020																	
																	ªÓ}ÎNw ÉsØ.>ž¿
00000040	98	17	BB	9D	75	B8	DE	07	C9	31	87	AB	D4	FD	2C	94	″.≫.u,Þ.Él‡«Ôý,″
00000050	1E	22	32	59	DD	81	6E	6B	A 8	64	12	3B	64	DC	37	10	."2YÝ.nk¨d.;dÜ7.
00000060	08	31	5A	83	F5	DF	37	74	38	4C	32	27	1E	4D	4B	6F	.1Zfőß7t8L2'.MKo
00000070	98	C1	EA	4B	6C	EE	55	5F	EC	09	32	5B	F2	AE	AO	FE	~ÁêKlîU_ì.2[ò⊗ þ
00000080	B8	98	BA	68	F4	72	E1	84	Fl	75	DA	EA	14	ЗD	35	DA	,~°hôrá"ñuÚê.=5Ú

Figure 6 - Encrypted PlugX shellcode in Hex editor.

The PlugX Malware loader decrypts and loads the encrypted shellcode (LMIGuardianDat.dat) inside the LMIGuardianSvc.exe. Injected memory space can be extracted to perform further analysis of decrypted PlugX Malware.

eneral Statistics	Performance Threads	Token Modules Memory	Environment Handles GP	U Disk and Network Comment			
✓ Hide free region	s					Strings Re	fresh
Base address	Туре		Size Protection	Use	Total WS P	rivate WS Shareable	WS
0x72d71000	Image: Commit		96 kB RX	C:\Windows\SysWOW64\ncrypt.dll	48 kB	48	8 kB
x72d51000	Image: Commit		80 kB RX	C:\Windows\SysWOW64\ncryptsslp.dll	36 kB	36	5 kB
x72d31000	Image: Commit		44 kB RX	C:\Windows\SysWOW64\NapiNSP.dll	16 kB	16	5 kB
x72d11000	Image: Commit		60 kB RX	C:\Windows\SysWOW64\pnrpnsp.dll	36 kB	36	5 kB
x72d01000	Image: Commit		40 kB RX	C:\Windows\SysWOW64\wshbth.dll	20 kB	20) kB
x72ce1000	Image: Commit		60 kB RX	C:\Windows\SysWOW64\nlaapi.dll	28 kB	28	3 kB
x72cd1000	Image: Commit		32 kB RX	C:\Windows\SysWOW64\winrnr.dll	12 kB	12	2 kB
x72c71000	Image: Commit		312 kB RX	C:\Windows\SysWOW64\FWPUCLNT.DLL	68 kB	68	8 kB
x10001000	Private: Commit		536 kB RX		536 kB	536 kB	
xb11000	Image: Commit		56 kB RX	C:\ProgramData\SamsungDriver\LMIGuardianSvc.exe	52 kB	52	2 kB
x4110000	Private: Commit		4 kB RWX		4 kB	4 kB	
x4100000	Private: Commit		4 kB RWX		4 kB	4 kB	
x1533000	Private: Commit		604 kB RWX	Injected PlugX Malware	604 kB	604 kB	
x3ffd000	Private: Commit		8 kB RW+G	Stack 32-bit (thread 5492)			
LMIGuardia	nSvc.exe (1840) (0x146	if000 - 0x1527000)				-	
00000c00 63	3 00 75 00 72 00	69 00 74 00 79 00 20	00 50 00 61 00 63 0	0 6b 00 61 00 67 00 65 00 00 00 00 00	c.u.r.i.t.yP.a.c.k.a.g.e		
				0 37 00 33 00 32 00 31 00 30 00 61 00			
				0 66 00 00 00 b3 d1 36 e0 00 09 00 88			
				0 30 00 61 00 35 00 30 00 34 00 36 00			
				0 00 0a 00 8c 06 00 00 00 59 00 00 00 0 22 00 00 00 67 00 00 00 68 00 00 00			
				0 22 00 00 00 67 00 00 00 68 00 00 00 00 00 00 00 00 00 00 00 00 00			
00000ca0 54							
00000ca0 54		00 00 66 00 00 00 22	00 00 00 67 00 00 0				
00000ca0 54 00000cc0 68 00000cc0 58	o 00 00 00 5e 00				[^f"ghk		
00000cc0 54 00000cc0 64 00000cc0 54 00000d00 98	o 00 00 00 5e 00 8 d1 2b e0 00 0c	00 88 4c 00 52 00 50	00 43 00 2d 00 38 0	00 68 00 00 00 66 00 00 00 00 00 00 00 00 00 37 00 33 00 32 00 31 00 30 00 61 00 00 66 00 00 00 9f d1 12 e0 00 0d 00 80	+L.R.P.C8.7.3.2.1.0.a.		
00000ca0 54 00000cc0 64 00000ce0 54 00000d00 98 00000d20 35	b 00 00 00 5e 00 3 d1 2b e0 00 0c 5 00 30 00 34 00	00 88 4c 00 52 00 50 36 00 64 00 64 00 61	00 43 00 2d 00 38 0 00 66 00 32 00 31 0	0 37 00 33 00 32 00 31 00 30 00 61 00	+L.R.P.C8.7.3.2.1.0.a.	C2 IF	•
00000ca0 54 00000cc0 68 00000cc0 58 00000d00 98 00000d20 35 00000d40 68 00000d40 68	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	00 88 4c 00 52 00 50 36 00 64 00 64 00 61 70 00 73 00 3a 00 2f 31 00 31 00 36 00 2f	00 43 00 2d 00 38 0 00 66 00 32 00 31 0 00 2f 00 32 00 31 0 00 2f 00 32 00 31 0 00 00 00 96 d1 15 6	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	+L.R.P.C8.7.3.2.1.0.a. 5.0.4.6.d.d.a.f.2.1.f. h.t.t.p.s.:///.2.1.71.22. 0.61.1.6./	C2 IF	•
00000ca0 54 00000cc0 68 00000cc0 58 00000d00 96 00000d20 35 00000d20 68 00000d20 68 00000d20 44	0 00 00 00 5e 00 3 d1 2b e0 00 0c 5 00 30 00 34 00 8 00 74 00 74 00 0 00 36 00 2e 00 4 03 00 00 ff ff	00 88 4c 00 52 00 50 36 00 64 00 64 00 61 70 00 73 00 3a 00 2f 31 00 31 00 36 00 2f ff ff 00 00 00 00 00	00 43 00 2d 00 38 0 00 66 00 32 00 31 0 00 2£ 00 32 00 31 0 00 2£ 00 32 00 31 0 00 02 00 32 00 31 0 00 00 00 96 d1 15 6 00 00 00 00 00 00 00 00	$ \begin{smallmatrix} 0 & 37 & 00 & 33 & 00 & 32 & 00 & 31 & 00 & 30 & 00 & 61 & 00 \\ 0 & 66 & 00 & 00 & 00 & 9f & d1 & 12 & e0 & 00 & 0d & 00 & 80 \\ 0 & 37 & 00 & 2e & 00 & 31 & 00 & 32 & 00 & 2e & 00 & 32 & 00 \\ 0 & 00 & 0e & 00 & 88 & 70 & ff & 46 & 01 & 80 & f5 & 4e & 01 \\ 0 & 00 & 0e & 00 & 00 & 00 & 00 & 00 $	+L.R.P.C8.7.3.2.1.0.a. 5.0.4.6.d.d.a.f.2.1.f. h.t.t.p.s::///2.1.71.22. 0.61.1.6./	C2 IF	•
00000ca0 54 00000cc0 58 00000ce0 58 00000d20 33 00000d20 33 00000d40 60 00000d40 30 00000d40 44 00000da0 00	b 00 00 00 5e 00 8 dl 2b e0 00 0c 5 00 30 00 34 00 8 00 74 00 74 00 0 00 36 00 2e 00 4 03 00 00 ff ff 0 00 00 00 00 00	00 88 4c 00 52 00 50 36 00 64 00 64 00 61 70 00 73 00 3a 00 2f 31 00 31 00 36 00 2f ff ff 00 00 00 00 00 00 00 00 8d d1 1c e0 00	00 43 00 2d 00 38 0 00 66 00 32 00 31 0 00 2f 00 32 00 31 0 00 2f 00 32 00 31 0 00 00 00 96 d1 15 6 00 00 00 00 00 00 00 00 0f 00 88 4c 00 52 0	$ \begin{smallmatrix} 0 & 37 & 00 & 33 & 00 & 32 & 00 & 31 & 00 & 30 & 00 & 61 & 00 \\ 0 & 66 & 00 & 00 & 00 & 9f & 41 & 12 & e0 & 00 & 00 & 00 \\ 0 & 37 & 00 & 2e & 00 & 31 & 00 & 32 & 00 & 2e & 00 & 32 & 00 \\ 0 & 00 & 0e & 00 & 88 & 70 & ff & 46 & 01 & 80 & f5 & 4e & 01 \\ 0 & 00 & 00 & 00 & 00 & 00 & 00 $. + L.R.P.C8.7.3.2.1.0.a. 5.0.4.6.d.d.a.f.2.1.f. h.t.t.p.s.://.2.1.71.22. 0.61.1.6. D	C2 IF	•
00000ca0 54 00000cc0 68 00000cc0 33 00000d20 33 00000d20 33 00000d20 34 00000d60 34 00000d80 40 00000d80 40	0 00 00 00 5e 00 8 11 2b e0 00 0c 5 00 30 00 34 00 8 01 74 00 74 00 8 00 36 00 2e 00 8 00 36 00 2e 00 4 03 00 00 00 00 0 00 00 00 00 00 4 00 36 00 31 00	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	$ \begin{smallmatrix} 0 & 37 & 00 & 33 & 00 & 32 & 00 & 31 & 00 & 30 & 00 & 61 & 00 \\ 0 & 66 & 00 & 00 & 00 & 9f & 41 & 12 & e0 & 00 & 00 & 00 \\ 0 & 37 & 00 & 2e & 00 & 31 & 00 & 32 & 00 & 2e & 00 & 32 & 00 \\ 0 & 00 & 0e & 00 & 88 & 70 & ff & 46 & 01 & 80 & f5 & 4e & 01 \\ 0 & 00 & 00 & 00 & 00 & 00 & 00 $		C2 IF	þ

Figure 7 – Memory map of LMIGuardianSvc.exe.

LMIGuardianDLL.dll (PlugX Loader) decrypts the LMIGuardianDAT.dat and loads it in memory of the legitimate process.

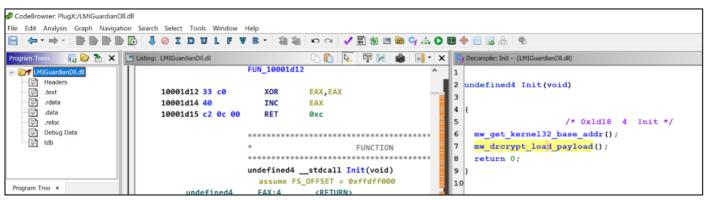


Figure 8 – Decompiled PlugX loader contains decryption function.

During static analysis, EclecticIQ analysts identified that the PlugX malware loader used a simple XOR algorithm to decrypt the LMIGuardianDAT.dat (XOR encrypted PlugX shellcode) to avoid signature-based detection from antimalware solutions.

1 🔤 Li	isting: LMIGuardianDII.dll - (12 addre	sses selected)	
	01 10	, , , , , , , , , , , , , , , , , , , ,	
	01 10	******	*******
		* 0x47F X0	DR KEY
		*****	**********
		0x47F XOR	
	100019fe 30 04 29	XOR	<pre>byte ptr [uVar8 + unaff_EBP*0x1],iVar5</pre>
	10001a01 <mark>41</mark>	INC	uVar8
	10001a02 <mark>3b cf</mark>	CMP	uVar8,EDI
+	10001a04 72 ec	JC	LAB_100019f2
0.0			n mx
	the second se		
_	ecompile: mw_drcrypt_load_payload	 (LMIGuardianD) 	ll.dll)
290			li.dli)
_			ii.dii)
290	if (uVar6 != 0) do {) {	(uVar6 >> 1);
290 291	if (uVar6 != 0) do { xor_key = :) {	(uVar6 >> 1);
290 291 292	if (uVar6 != 0) do { xor_key = :) <mark>{</mark> xor_key + /* 0x47F X	(uVar6 >> 1); OR KEY */
290 291 292 293 294	<pre>if (uVar6 != 0) do { xor_key = ; pcVar1[uVa;</pre>) { xor_key + /* 0x47F X r3] = (cod	(uVar6 >> 1);
290 291 292 293 294 295	<pre>if (uVar6 != 0)</pre>) { xor_key + /* 0x47F X r3] = (cod ar3 + 1;	<pre>(uVar6 >> 1); OR KEY */ e)((byte)pcVar1[uVar3] ^ (byte)xor_key);</pre>
290 291 292 293 294	<pre>if (uVar6 != 0) do { xor_key = ; pcVar1[uVa; uVar3 = uV; } while (uVa;</pre>) { xor_key + /* 0x47F X r3] = (cod ar3 + 1;	<pre>(uVar6 >> 1); OR KEY */ e)((byte)pcVar1[uVar3] ^ (byte)xor_key);</pre>

Figure 9 – XOR key is stored statically to perform decryption during execution time of PlugX loader.

PlugX loader used a static XOR key "0x47F", to decrypt the PlugX shellcode. The below image shows a Python script being used to decrypt the LMIGuardianDAT.dat.

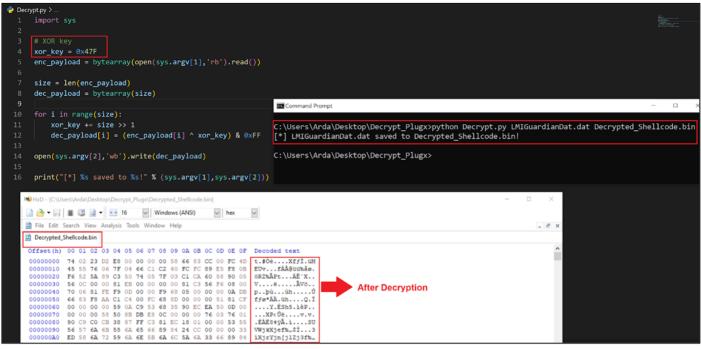


Figure 10 – Decrypted PlugX shellcode.

Once the PlugX malware has been executed in-memory, the C2 config is decrypted. The C2 IP address 217[.]12[.]206[.]116 and the campaign ID of "test2022" are seen in the figures below:

Listing	g: 1000000	0.dll		h 🛍	💊 🖳 📝	🐞 📑 🗸 🔀 Decompile: get_campaign_ID - (10000000.dll)
10		00	73	 campaign_ID unicode	u"test2022"	<pre> 1 2wchar_t * get_campaign_ID(void) 3 4{ 5 return campaign_ID; </pre>
10	0095e16	00		??	00h	6 }

Figure 11 – Decompiled PlugX malware contains campaign ID as a fingerprint of the attack to categorize the victims.

	Listing: 1000000	0.dll					• • ×		C Decompile: get_C2_IP - (10000000.dll)
				(C2_IP		<mark>^</mark>	1	undefined * get_C2_IP(void)
÷	1009608c	01			??	01h		3	3
	1009608d	00			55	00h	-	4	
	1009608e	bb			22	BBh	-	5	5 return & C2 IP;
	1009608 f	01			55	01h	-	6	5 }
	10096090	32	31	37	ds	"217.12.206.116"		7	7
		2e	31	32					
		2e	32	30					

Figure 12 - Decompiled PlugX malware contains command and control (C2) IP address as static.

Third Stage: Registry Run Key Persistence

Mustang Panda abuses Windows registry run keys to gain persistence on the infected system. On Windows operating systems the run registry keys execute the specified program when a user logs on to the device.

The PlugX malware created a new run key called as LMIGuardian Update, shown in the image below.

Process Name test2022.ucp test2022.ucp test2022.ucp test2022.ucp test2022.ucp test2022.ucp	732 🔮 Reg 732 🔮 Reg 732 💕 Reg	ation gCreateKey gSetInfoKey gQueryKey gSetValue gCloseKey	Path HKCU\Software\Microsoft\Windows\CurrentVersion\Run HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\LMIGuardian Update HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run	Result SUCCESS SUCCESS SUCCESS SUCCESS SUCCESS
	Event Properties Event Date: Thread: Class: Operation: Result: Path: Duration: Type: Length: Data:	Process Stack 12/30/2022 8:16:46:2857801 AM 8984 Registry RegSetValue SUCCESS HKCU\SOFTWARE\Microsoft\W 0.0000514 REG_1 134	indows\CurrentVersion\Run\LMIGuardian Update	

Figure 13 – Persistence established by malware after writing a new Run key.

Every logon will cause the Windows registry run key to execute the LMIGuardianSvc.exe, triggering the DLL Hijacking that leads to PlugX malware execution.

nputer\HKE	Y_CURRENT_USER\S	SOFTWARE\Microsoft\Window	<pre>s\CurrentVersion\</pre>	Run
RADAR	۲ ۲	Name	Туре	Data
🗙 🕨 Run		el(Default)	REG SZ	(value not set)
Auto RunOn		MLMIGuardian Update	REG_SZ	"C:\Users\RE\AppData\Roaming\SamsungDriver\LMIGuardianSvc.exe" 7
> 📕 Screen	savers	Edit String	×	
Search Search	Settings	Value name: LMIGuardian Update		
> 📕 Securit > 📕 Setting	ty and Maintenan gSvnc	Value data: C:\Users\RE\AppData\Roaming\SamsungDrive	r\LMIGuardianSvc.exe* 748	
-	xtensions		OK Cancel	

Figure 14 – Written registry key.

The malware creates a new file path which is being used by the persistence mechanism (Run key) to execute the LMIGuardianSvc.exe on this specific file path:

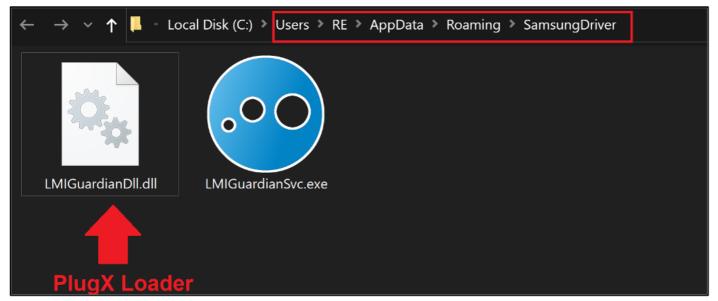


Figure 15 – New file path created for persistence execution of PlugX malware.

Fourth Stage: Command and Control Connection

After a successful execution of PlugX malware, it connects to a remote C2 server which is used to send commands to compromised systems via the PlugX malware and to receive exfiltrated data from a target network.

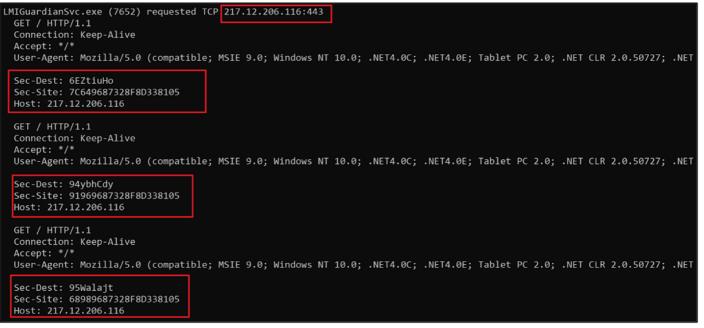


Figure 16 – Request headers and server response observed in Mustang Panda's customized PlugX variant.

Once the device is infected, an attacker can remotely execute several kinds of commands on the affected system. 'Sec-Dest' and 'Sec-Site' HTTP sections contain encrypted data of victim machine information sent to attackers.



Figure 17 - Network capture during the TCP request to remote C2 server over port 443.

The C2 IP address 217[.]12[.]206[.]116 was seen hosting another service on port 8088 with a unique SSL certificate that is itself issued to the IP address 45[.]134[.]83[.]29, which is identified as additional Mustang Panda's infrastructure, according to the BlackBerry Research & Intelligence Team [1].

S 217.12.206.116 × +	
← → C ▲ Not secure https://217.12.206.116:8088	
Certificate ×	Ē.
	This page isn't working
Issued to: 45.134.83.29	217.12.206.116 didn't send any data.
Issued by: CTA Root CA	ERR_EMPTY_RESPONSE
Valid from 2022-02-23 to 2032-03-03	Reload

Figure 18 – Issued SSL certificate contains another IP address, which was used by Mustang Panda APT group for previous attacks. [1]

Conclusion

EclecticIQ analysts assess it is almost certain the APT group Musta Panda was responsible for this attack. Mustang Panda has leveraged PlugX malware in previous campaigns targeting the Ukraine and has used similar TTPs like DLL hijacking. The group previously used Windows shortcut (LNK) files disguised using double extensions (such as .doc.Ink) with a Microsoft Word icon and has abused registry run keys for persistence. The SSL certificated used in this attack overlaps with previous Mustang Panda activity targeting the Ukraine.

□ Name	Date modified	Туре	Size	
	10/25/2022 5:36 AM	File folder		
Political Guidance for the new EU approach towa	rds Russia.doc 10/25/2022 5:36 AM	Shortcut		3 KB

Figure 19 – Example of LNK Phishing lure used by Mustang Panda APT group in their previous attacks. [2]

EclecticIQ analysts assess it is probable the target for this lure document was a European entity. The phishing lure used in the campaign discusses the effect EU sanctions against Russia will have on the European Union. Mustang Panda has targeted European organizations before in a similar campaign in 2022-10-26 [Figure 19]. Mustang Panda APT group continues to be a highly active threat group conducting cyber operations targeting organizations across Europe [2]. EclecticIQ analysts have identified Mustang Panda operators adding new evasion techniques, like using a custom malware loader to execute

encrypted an PlugX sample for the purpose of increasing infection rates and staying under the radar while performing cyber espionage activates against victims.

EclecticIQ analysts assess that it is probable Mustang Panda will increase their activity and continue to use similar TTPs in response to geopolitical developments in Ukraine and Europe, based on an examination of the group's previous cyberespionage activity. Analysts should continue to track Mustang Panda using the TTPs and infrastructure highlighted in the report and the YARA rules provided below.

Mitigations

- Implement basic incident response and detection deployments and controls like network IDS, netflow collection, host-logging, and web proxy, alongside human monitoring of detection sources.
- Employ host-based controls.
- Filter email correspondence and monitor for malicious attachments.
- Identify critical data and implement additional network segmentation and special protections for sensitive information, such as multifactor authentication, highly restricted access, and storage systems only accessible via an internal network.
- Create alerts for disk image file types, such as ISO, and shortcut files, which have been increasingly abused by different threat actors. Furthermore, organizations should consider disabling auto-mounting of ISO or VHD files.
- Configure intrusion detection systems (IDS), intrusion prevention systems (IPS), or any network defence mechanisms in place to alert on and upon review, consider blocking connection attempts from unrecognized external IP addresses and domains.

MITRE ATT&CK

Tactic: Technique	ATT&CK Code
Execution: User Execution Malicious File	T1204
Defense Evasion: Hijack Execution Flow DLL Search Order Hijacking	T1574.001
Defense Evasion: Deobfuscate/Decode Files or Information	T1140
Defense Evasion: Masquerading Double File Extension	T1036.007
Command-and-Control: Encrypted Channel Symmetric Cryptography	T1573.001
Command-and-Control: Data Encoding Standard Encoding	T1132.001
Persistence: Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	T1547.001

INDICATORS OF COMPROMISE

Sample File
Name(s)SHA-256 HashLMIGuardianDII.dllee2c8909089f53aafc421d9853c01856b0a9015eba12aa0382e98417d28aef3fLMIGuardianDat.dad8c4926dd32204b6a666b274a78ccfb16fe84bbd7d6bc218a5310970c4c5d9450draft letter to
European
Commission
RUSSIAN OIL723d804cfc334cad788f86c39c7fb58b42f452a72191f7f39400cf05d980b4f3

PRICE CAP sg de.iso draft letter to European Commission RUSSIAN OIL PRICE CAP sg de.doc.lnk LMIGuardianSvc.exe renamed (test2022.ucp) Command and Control Servers

217[.]12[.]206[.]116 45[.]134[.]83[.]29

Hunting Resources: Live Queries & Yara Rules