Operation ‘Kremlin’ – ClearSky Cyber Security

Published: 2021-01-07 · Archived: 2026-04-29 07:44:07 UTC

Introduction

ClearSky researchers identified a malicious “.docx” file that was uploaded to VirusTotal from Russia in mid-December. The file contains an obfuscated URL to a remote template which contains malicious VBA, eventually leading to the execution of VBS on the infected machine. **The attack’s purpose is to stealthily exfiltrate information without running any external executables on the system**.

Notably, the process is escalated on a certain day of the week, suggesting a possible familiarity with the intended victim or victims.

We estimate with medium confidence that the same threat actor responsible for the attacks described in this paper also conducted an attack named “Operation Domino[1]” that occurred earlier in 2020.

We decided to name the operation “Kremlin” due to the use of a parameter named “kreml” in the “poslai” (meaning send in Russian) function that exfiltrates the data.

TTPs

The lure document contains a “bonus payment” request from **The Ministry of Defense of the Russian Federation**, to be submitted by ministry workers for the period of November 2020.

The document was uploaded from Russia to VirusTotal on the 16-12-2020, and the date that appears at the end of the document is 30-11-2020. The document was last modified at 01-12-2020.

The malicious file contains an embedded XLS file, which is displayed when the document is opened[2]. This file is not detected as malicious on VirusTotal.

Template Injection

The docx file abuses remote template injection. The adversary is using an interesting method to hide the URL to the malicious file. Typical abuse would include adding a remote code IP address, as can be seen at Trend Micro's image[3]:

However, instead of writing the IP address in dot-decimal notation[4] the adversary used the integer representation. 1587326585 translates[5] to the IP address 94.156.174.121

The remote file is a DOTM, and was uploaded to VirusTotal from Russia as well: virustotal.com/gui/file/56f5cb1590912dc6dfa0945f4d6e49500f238b5d4847ab3da24c7f848c12217d/details

VBS

This DOTM file contains VBA that writes a VBS file and executes it:

The created VBS file is:
virustotal.com/gui/file/e72670493c5cbccecf7028fdfad4e166bf0b51cc4d41cb2ac85a59f08ccd2627/details

The VBS begins its activity by creating an array with the full paths to the “Desktop” and “My Documents” special folders, simultaneously creating a “rec” variable with the path of the recently opened file’s special folder in windows.

Then, the VBS calls psr.exe[6]. As far as we know, this is the first documented use case of psr.exe as LOLBIN[7] by an APT group.

The VBS has a specific check for the day of the week the file is opened on:

If the file is opened on a Wednesday, the VBS will look up a key in the registry and will query all system drives. If the drive type is not 4 (CD/DVD), the VBS will continue. Afterwards the VBS checks if the drive letter is other than “A” (Thank you @jaydinbas for the correction).

If both checks on the queried drive pass, the VBS will look for folders that start with “PROGRA” in their name and pass the folder name to a procedure named “subse”, this includes remote drives, if attached. The “subse” procedure code is the following:

It checks if the folder name starts with “WINDOW”. If it does, the code gets the file names in the folder and each file’s extension is checked via the “repl” function:

The “repl” function checks whether the file’s extension starts with 2 specific characters from a list of such pairs. This is most likely done to avoid suspicious full file extensions while filtering the interesting files. From the first 2 characters we can guess which files are of interest:

2 First Characters	Signification
DO	DOC/DOCX MS Word files
XL	XLS/XLSB MS Excel files
PP	PPT/PPTX MS Power Point files
RT	RTF Rich Text Format files
ZI	Zip Archives
TX	TXT Plain Text Files
CS	CSV Comma Separated Values

If a file with an interesting extension exists, the filename is sent as parameter for the “sendme” procedure.

The “sendme” function **assumes that WinRAR exists on the running computer**, and it uses “rar.exe” to compress and password protect the file of interest.

After compression, the file is loaded into a buffer, if the buffer is too big, nothing happens. This is most likely done to avoid big zip files.

However, if the buffer is within the range the adversary specified, it will encode the bytes with base64 using the “enco” function:

The base64 encoded bytes are then sent to the “poslai” procedure:

As a side note, many of the function and variable names are in the Russian language, the word “poslai” means “send”. The word “kreml” is the Russian word for “Kremlin”[9]. This is why we decided to call this paper “Operation Kremlin”.

The “poslai” randomly chooses between appending 2 seemingly random integers to the string “http://”:

As we have seen in the DOC file, 1587326585 translates[10] to the IP address 94.156.174[.]121, while 3119738898 translates to 185.243.112[.]18

Next, there is a “for” loop that takes the currently logged-on username and converts every character into its decimal representation, summing the values.

If the username is admin, then: **a=97+d=100+m=109+i=105+n=110=>521**

The “now()” function in VBS returns the current system time in the following format:

After encoding the currently logged on user, there is an “encoding” of the current system time if the file size is bigger than 0:

If the size is bigger than zero, the current system time is formatted to remove special characters such as ":" "." "/". Then "12435687" is appended to the formatted string, and only the first 14 characters are used.

If we take the time from the MsgBox image of "now()" we captured, the formatting before the concatenation would result in "1228202073443PM", a 15 character long string. If the time was set to 24h instead of 12h with AM/PM representation the string would be exactly 14 characters long:

Finally, there is a "do while" loop, that truncates the archived file of interest that was base64 encoded, into chunks of 942 characters that are saved into the "tsip" variable. After 942 characters have been acquired, it appends them to the "uri" variable and sends a GET request to the server.

However, if the size of the file is zero it will send a get request to a file named “patch.png”. If the request’s status code is 200 and the length of the response is bigger than 13, the VBS will write the response to the registry, avoiding writing potentially malicious code directly to the disk.

Next, the VBS creates a new VBS file named “diagnostics.vbs” in the Windows StartUp special folder. This new VBS reads the registry key that has been written from the http response. Since the VBS is in the StartUp folder, this seem to be a persistence mechanism.

If the response is 200 but the length is less than 13, the VBS will launch “msiexec” silently with the URL http://5.9.242[.]126/521/patch.gif (521 in case the username was admin). All of this is done specifically if the file is run on a Wednesday. Additionally, there is a piece of code that runs regardless of the day of the week:

The VBS checks if a file named “myphone_diag.log” exists on the system, if it does the code above won’t execute. If the file doesn’t exist, the VBS will start doing something similar to what happens in case of a Wednesday activation, but for local folders only.

Once again, the purpose of the code is stealthy data exfiltration, it starts with the recently opened file’s folder, listing all the LNK files and sending them to the “ExTarPa” procedure:

The procedure extracts the target path of the actual file that was recently opened, checks the extension against the list of interesting extensions as previously shown in the code of the “repl” function.

Once finished, the VBS proceeds to do the same data exfiltration scheme for the “Desktop” and “My Documents” folders. After the check for the file “myphone_diag.log”, there is the following piece of code:

This code sleeps for 5 seconds and stops psr.exe. Following this, the VBS performs a WMI query to get the installed software and its version on the computer:

Later, the VBS uses the priorly described “sendme” procedure to exfiltrate those 2 files. As seen in some of the code above, there are many http connections if there are interesting files found, but all of them returned 404 status code from the server. However, it’s a trap:

As we described through this post, the data is actually being exfiltrated as part of the URI. The adversary has some sort of control over the C2 servers, allowing him to read the access logs, reconstructing the exfiltrated archive files from them, without actually sending any data in traditional ways.

This entire data exfiltration scheme is conducted stealthily, without a single malicious executable file being downloaded to the computer.

Attribution

The VBS communicates with two IP addresses that were previously observed[11] in an attack leveraging previously unseen 1-day exploitation of CVE-2020-0968. This was called “Operation Domino[12]” by a Chinese security firm.

The URL we observed as part of “Operation Domino” was: hxxp://94.156.174[.]7/up/a1a.htm.

In the Chinese report, they successfully ran the dropped DLL file which communicated with the IP address 185.243.112[.]57. There is a clear connection by the infrastructure used by the adversary:

Operation “Domino”	Operation “Kremlin”
СВЕДЕНИЯ О ПОДСУДИМОМ.rtf	Tabel_premia_N20.docx
94.156.174[.]7	94.156.174[.]121
185.243.112[.]57	185.243.112[.]18

In both cases the URI contained the word “up”. Another strong connection between the two files is the unique language set, Russian and Arabic from Saudi Arabia. Both attacks are complex and carefully tailored for Russian speaking targets, using very unique attack techniques that are not seen in wide use.

We estimate with medium confidence that both attacks are from the same threat actor. At this point we can’t attribute this to a specific known threat actor, however, during the analysis of “Tabel_premia_N20.docx” we have managed to trace another attack by the same threat actor.

2019 Attack

This attack utilizes a very similar VBS code, but the attack vector is different, so the analysis will mainly focus on the attack vector as the VBS code should be almost the same as described in the main section above. We have observed the following html file:

57f0252b8d2a7d946ec2231c546986728c1141ceba95f7a5128a40796b928519

This file was uploaded from Russia at 2019-04-15 to VirusTotal. We assume it has been sent as an attachment to an email.

The logo in the HTML page is of Inter Raoues[13], a Russian energy company. As can be seen in the image above, the "footer.png" image didn't load, but we believe it was never meant to load anything, except give the adversary an indication someone accessed the HTML file:

This HTML file automatically "downloads" an HTA file with the following hash:

a7091e1c532351ae33a8d51523a7b5cc708bd8299cb87951a2c52fe816da90a3

This file was uploaded to VirusTotal from Russia on 2019-04-16. The HTA writes to a unique registry key that we have observed in the 2020 attack:

Then the HTA creates a VBS file that reads the contents from the registry and executes it, a similar technique was observed in 2020 with the "diagnostics.vbs" file. The resulting VBS file hash is:

2c13aa6c2240166ddb1b5b9b22e3868ea9c094424c5056b9f557a7fa906ce564

This file was uploaded to VirusTotal from Russia on 2019-04-23.

We have decoded the values from the registry and uploaded the resulting VBS file to VirusTotal: virustotal.com/gui/file/557af395b358bd787402e6b1827c7c69e41bedc43e95e35529268264f564866b/detection

Some of the function names, as well as the variables, are the same as in the VBS file from 2020, like "poslai(kreml)".

Once again, there is a use of the word "up" in the C2 URI, the extension is PHP as well. The C2 are different and accessed via domain name and not an IP, which doesn't correlate to what was observed in 2020:

One of the C2 servers is the domain that was observed from the missing footer image, bibigreen[.]ru. The 2nd C2 is hesheflowershop[.]ru.

The 2019 VBS also uses “psr.exe” to spy on the victim. The 2019 VBS has the same, very specific check for Wednesday, and if there is a drive with the letter A connected to the system on that day:

There are many differences between the 2019 VBS & the 2020 VBS, but the essence of the code is the same, exfiltrate confidential information from the affected system without running any 3rd party external executable on the system.

Epilogue

Since we have seen IPs form two sets of pairs coming from the same subnets, we highly recommend monitoring any network connections with the following subnets:

94.156.174.0/24

185.243.112.0/24

We have found another connection between the infrastructure used in 2020 by “Operation Kremlin”.

The connection leads to the same person “Edvinas Vyzas” and to the same building in Seychelles, however the phone number is associated with New Zealand.

Indicators

MD5	filename	Description
f745d6e3c811c9c06acb2ebc45a174ba	Tabel_premia_N20.docx	2020 initial lure document
5dd05f94ebdeb7afb494b541b317eb8c	_____Microsoft_Excel_97-20031.xls	2020 embedded table in initial lure document
60cb0e31510a9cd747daae323d28f489	Dc2.dotm	2020 remote template containing malicious VBA
0a62afe0dfe369b5280c432533671aa0	backup.vbs	2020 VBS payload
fb848fcf49054871f62bbf9b5f5c9282	zanpoc_energy.html	2019 initial lure document
669190300c47c141c35ea3867061512e	ENERGY_PRODACTION.html.hta	2019 malicious HTA

54a4302989bf0f3ae42d2aeddca50e1a	backup.vbs	2019 minimal VBS payload
3e38a347b3914893cb5fb92d12558003	diagnostics.vbs	2019 VBS payload extracted from registry

Network Address	URI Path	Network Name (RIPE)
Bibigreen[.]ru	/wp-content/energia/wp/	REGRU-NETWORK
Bibigreen[.]ru	/up/up.php	REGRU-NETWORK
hesheflowershop[.]ru	/wp/up.php	REGRU-NETWORK
94.156.174[.]121	/wp/521/patch.png	NETERRA-CINFUCOM-NET
94.156.174[.]121	/wp/521up.php	NETERRA-CINFUCOM-NET
185.243.112[.]18	/wp/521up.php	CrownCloud
5.9.242[.]126	/521/patch.gif	CLOSCO-LTD

The value “521” in the URI Path can be different, depending on the executing user, as described in this blog.

---

[1] ti.dbappsecurity.com.cn/blog/index.php/2020/09/18/operation-domino/

[2] virustotal.com/gui/file/907ff4964ec8cdb1a8e5ac6005dd74aca7bd01c941f4c4bfff0c1a03dd695f83/details

[3] trendmicro.com/en_us/research/17/h/cve-2017-0199-new-malware-abuses-powerpoint-slide-show.html

[4] en.wikipedia.org/wiki/Dot-decimal_notation

[5] vultr.com/resources/ipv4-converter/?ip_address=94.156.174.121

[6] cyberarms.wordpress.com/2016/02/13/using-problem-steps-recorder-psr-remotely-with-metasploit/

[7] lolbas-project.github.io/lolbas/Binaries/Psr/

[8] docs.microsoft.com/en-us/dotnet/api/system.io.drivetype?view=net-5.0

[9] *“The name “Kremlin” means “fortress inside a city”, and is often also used metonymically to refer to the government of the Russian Federation in a similar sense to how “White House” refers to the Executive Office of the President of the United States”*

[10] vultr.com/resources/ipv4-converter/?ip_address=94.156.174.121

[11] docs.google.com/document/d/1oYX3uN6KxIX_StzTH0s0yFNNoHDnV8VgmVqU5WoeErc/edit#heading=h.i0mazx2vmwtx

[12] ti.dbappsecurity.com.cn/blog/index.php/2020/09/18/operation-domino/

[13] en.wikipedia.org/wiki/Inter_RAO
